Verification of Graph Algorithms in the Context of Wireless Networks

by

Lucas Maximilian Böltz

approved dissertation for the academic degree of
Doctor of Natural Sciences (Dr. rer. nat.)

Faculty 4: Computer Science
of the University of Koblenz

Chair of doctoral committee: Prof. Dr. Ralf Lämmel

Reviewers: Prof. Dr. Viorica Sofronie-Stokkermans, Prof. Dr. Hannes Frey

Chair of doctoral commission: Prof. Dr. Stefan Müller

Date of disputation: 10.06.2026

Summary

The goal of this thesis is to use methods from automated reasoning to analyze graphs occurring in the context of wireless networks. For this a suitable representation of the considered graphs, the transformations and algorithms executed on the graphs and also the underlying geometric conditions are required. The considered graphs resp. classes of graphs in this thesis refer to a) graphs only defined by geometric conditions, b) graphs defined by more general axioms after applying transformations, c) graphs resulting from a graph transformation algorithm.

For studying graphs in the context of wireless networks, an important task is the construction of graphs without intersecting edges. Such graphs are called plane graphs and have a wide range of applications, for example in routing procedures. For the construction of connected plane graphs on a given set of vertices, the concepts of Gabriel graphs and relative neighborhood graphs exist. These concepts were originally developed for the Euclidean plane with Euclidean distance. In this thesis it is investigated, for which more general distance functions, Gabriel graphs, and relative neighborhood graphs are plane graphs, whenever possible by using the hierarchical prover HPiLoT.

In addition to graph constructions, this thesis also examines graph transformations. A transformation is simple if the existence of an edge uv in the transformed graph depends only on the existence of the edges uv and vu in the underlying original graph. In the original graph, predicates determine which edges must exist, which edges must not exist, and which edges exist depending on the existence of other edges. It is investigated under which conditions for the predicates and for which transformations expressions can be found to automatically examine the containedness relation between two graph classes, whereby only graphs up to a certain number of vertices need to be considered. For the expressions found in this way, the containedness relations are evaluated for all possible graph classes using Mathematica.

Graphs can also be transformed using algorithms instead of simple transformations. This thesis examines a planarization algorithm that generates a plane graph from an original graph by selecting a subset of edges. To maintain, in addition also the graph's connectivity, the original graph must satisfy certain structural properties. Such properties are the redundancy and coexistence property. It is shown that for RCGs, i.e. graphs that satisfy these two properties, a connected plane subgraph always exists. To obtain this connected plane graph, the correctness of a local planarization algorithm for RCGs is proven by verifying the individual proof steps of a manual proof using the theorem prover Z3. To perform this verification, a suitable axiom system is defined that describes, the properties of the graphs, the properties required for the proof, as well as the underlying geometry.

Zusammenfassung

Im Rahmen dieser Dissertation wird die Möglichkeit untersucht, Methoden des automatischen Beweisens für Graphen im Kontext von drahtlosen Netzwerken zu verwenden. Dafür müssen die Graphen, die betrachteten Transformationen bzw. Algorithmen, sowie geometrischen Eigenschaften, in einer geeigneten Weise beschrieben werden. Im Rahmen dieser Dissertation wurden a) rein geometrisch definierte Graphen, b) durch allgemeinere Axiome definierten Graphen nach dem anwenden einer simplen Transformation, sowie c) durch Graphtransformationsalgorithmen erzeugte Graphen untersucht.

Eine wichtiges Ziel bei der Betrachtung von Graphen im Kontext von drahtlosen Netzwerken ist, die Konstruktion von Graphen ohne schneidene Kanten. Solche Graphen werden als ebene Graphen bezeichnet und haben vielfältige Anwendungsgebiete, zum Beispiel für Routingverfahren. Für die Konstruktion von zusammenhängenden ebenen Graphen auf einer gegebenen Knotenmengen existieren die Konzepte Gabriel-Graph und relativer Nachbarschaftsgraph. Diese ursprünglich für die euklidische Distanz konzipierten Graphen werden im Rahmen der Dissertation für allgemeineren Distanzfunktionen analysiert und dabei wird, sofern möglich, der hierarschische Beweiser HPiLoT verwendet.

Neben Graphkonstruktionen werden in der Dissertation auch Transformationen auf Graphen betrachtet. Eine Transformation ist simpel, wenn die Existenz einer Kante uv im transformierten Graphen, nur von der Existenz der Kanten uv und vu im zugrundeliegenden ursprünglichen Graphen abhängt. Bedingungen für die Existenz von Kanten im zugrundeliegenden Graphen, werden mithilfe von Prädikaten axiomatisch beschrieben. Es wird untersucht, unter welchen Bedingungen für die Prädikate und für welche Transformationen Ausdrücke gefunden werden können, mit denen die Teilklassenbeziehung zwischen zwei Graphklassen automatisch untersucht werden kann, wobei nur Graphen bis zu einer gewissen Anzahl an Knoten betrachtet werden müssen. Für die so gefundenen Ausdrücke wird die Teilklassenbeziehung für alle möglichen Graphklassen mittels Mathematica ausgewertet.

Eine weitere Möglichkeit Graphen in andere Graphen zu transformieren sind Algorithmen. Im Rahmen der Dissertation wird ein Planarisierungsalgorithmus, der aus einem Ursprungsgraphen einen ebenen Graphen durch Auswahl einer Teilmenge von Kanten erzeugt, betrachtet. Um gleichzeitig den Zusammenhang des Graphen beizubehalten, muss der Ursprungsgraph, gewisse strukturelle Eigenschaften erfüllen, z. B. die Redundanz- und Koexistenzeigenschaft. Es wird gezeigt, dass für Graphen, welche die Redundanz- und Koexistenzeigenschaft erfüllen, immer ein zusammenhängender ebener Teilgraph existiert. Um diesen zusammenhängenden ebenen Graphen zu erhalten, wird die Korrektheit eines lokalen Planarisierungsalgorithmus durch die Verifizierung der einzelnen Beweisschritte eines manuellen Beweises mit dem Theorembeweiser Z3 gezeigt. Um diese Verifizierung durchzuführen, wird zunächst ein geeignetes Axiomensystem definiert, das sowohl die zugrundeliegende Geometrie als auch die Eigenschaften der Graphen und für den Beweis benötigte Eigenschaften beschreibt.

Acknowledgment

I would like to thank my advisors, Prof. Dr. Hannes Frey and Prof. Dr. Viorica Sofronie-Stokkermans, for their supervision and guidance over the past few years. Both supported my research and this thesis would not have been possible without their support. I would also like to thank Dr. Dennis Peuter and Dr. Steffen Böhmer for their collaboration and the helpful discussions in our joint work. Last but not least, I want to thank all family members and friends who have supported me throughout my life.

Contents

1 Introduction **1**
- 1.1 Graphs in Wireless Sensor Networks . . . 2
- 1.2 Application of Plane Graph Drawings . . . 2
- 1.3 Transformations on and Containedness Between Graph Classes . . . 3
- 1.4 Using Automated Reasoning Approaches for Geometric Graphs . . . 5
- 1.5 Contribution of the Thesis . . . 7
- 1.6 Publications . . . 10
- 1.7 Structure of the Thesis . . . 12

2 Preliminaries **13**
- 2.1 First-order Logic . . . 13
- 2.2 Theories and Local Theory Extensions . . . 15
  - 2.2.1 Local Theory Extensions . . . 16
  - 2.2.2 Hierarchical Reasoning in Local Theory Extensions . . . 17
- 2.3 Graph Models and Algorithms in Wireless Networks . . . 18
  - 2.3.1 Basic Graph Classes in Wireless Network Models . . . 20
  - 2.3.2 Connectivity and Planarity of Graphs . . . 22
  - 2.3.3 Graph Transformations and Graph Constructions . . . 24
  - 2.3.4 Locality in Graphs . . . 26
- 2.4 Provers . . . 27

3 Graph Class Containedness Relations Between Graph Classes Related to Planarity Conditions **29**
- 3.1 Graph Classes Related to Planarity Conditions . . . 29
  - 3.1.1 Axiomatic Description of the Graph Classes . . . 29
  - 3.1.2 Conditions for Testing Class Containedness Relations . . . 31
- 3.2 Locality of Theory Extensions Involving Distances . . . 31
- 3.3 Class Containedness Relations for Different Distance Functions . . . 33
  - 3.3.1 Euclidean Distance and Arbitrary Metric . . . 34
  - 3.3.2 Manhattan and Maximum Metric . . . 35
  - 3.3.3 Distances Induced by Arbitrary p-norms . . . 36
  - 3.3.4 Distances Induced by Quasi-norms . . . 40
  - 3.3.5 Overview about the containedness relations for all considered distance functions . . . 43

4 Graph Class Containedness Relations Between Graph Classes Obtained by Simple Transformations **45**
- 4.1 Axiomatic Description of Graph Classes . . . 45

4.2 Simple Graph Class Transformations 47
4.3 Proving Graph Class Containedness Relations 49
4.3.1 General Properties 49
4.3.2 Properties for the Graph Classes Obtained by Transformations in Γ_1 52
4.3.3 Axiomatization for the Graph Classes Obtained by Transformations in Γ_2 59
4.3.4 Axiomatization for the Graph Classes Obtained by Transformations in Γ_3 66
4.3.5 Expressions for Proving Subclass Relations 69
4.4 Example Application 77
4.4.1 Manual Application of the Derived Concept 77
4.4.2 Automatically Derived Graph Class Containedness Relations 78

5 Construction of Connected and Plane Subgraphs 85
5.1 Connected Planarization Under Acyclic Redundancy Property 85
5.2 Connected Planarization for RCGs 89
5.3 The Connected Planarization Algorithm for RCGs 97
5.3.1 Algorithm Description 98
5.3.2 Local Variant of the CP-Algorithm 100

6 Modelling Geometric Graphs 101
6.1 Reasoning in Geometry and Graph Theory 101
6.2 An Axiomatic system for RCGs and the CP-algorithm 102
6.2.1 Geometric Axioms 103
6.2.2 Axiomatizing Graph Properties 109
6.2.3 Axioms for Notions Used in the CP-algorithm 109
6.2.4 Axioms for Notions Used in the Correctness Proof 110
6.3 Analyzing the Axioms for the Verification 112
6.3.1 Decidability in the Proposed Axiom System 112
6.3.2 Choosing the Proof Systems 114
6.3.3 Automated Proof of Properties of Geometric Graphs 114

7 Correctness Proof of the CP-algorithm 119
7.1 Idea of the Correctness Proof 119
7.2 Removed Edges are Intersected by Selected Edges (Lemma 20) 121
7.2.1 Proof of Part1: Intersection of the Eligible Edges with u_1v_1 130
7.2.2 Proof of Part2: No Redundancy Cycle Formed by Incident Edges . . 139
7.2.3 Proof of Part3: No Redundancy Cycle Among Switching Edges . . . 153
7.3 Connection Between the Components X and Y 180
7.3.1 Acylicity of the Sequence of Connecting Edges (Lemma 21) 180
7.3.2 Path connecting the components X and Y (Lemma 22) 181
7.4 Constructing of Counterexample Under Weaker Conditions 184

7.5 Details on the Tests 187

8 Conclusion **193**

9 Appendix **205**

9.1 Automatically Derived Containedness Relations 205
9.2 Automatically Derived Counter Examples 213
9.3 Overview about the axioms used for the verification of the CP-algorithm . . 216
9.3.1 Axioms for left 216
9.3.2 Axioms for intersection 217
9.3.3 Axioms for inside 217
9.3.4 Axioms for edges E in the input graph of the algorithm 218
9.3.5 Axioms for edges F in the output graph of the algorithm 218
9.3.6 Axioms for the convex hull of the vertices in $\Delta u_1 p_1 w_1$ 219
9.4 Proof tasks 220
9.4.1 Proof tasks of closestF 220
9.4.2 Proof tasks of orderF 221
9.4.3 Proof tasks of Part1 222
9.4.4 Proof tasks of Part2 244
9.4.5 Proof tasks of Part3 256
9.4.6 Proof tasks of part4 328
9.4.7 Proof tasks of part5 329

1 Introduction

In this thesis different approaches for using automated reasoning for analyzing graphs occurring in wireless sensor networks are studied. The graphs studied in this context are typically geometric graphs with vertices represented as points in the Euclidean plane and edges between the vertices are drawn as straight-line segments. While the positions of the vertices are often randomly distributed, but fixed once they are set, the existence of edges between the vertices depends on geometric conditions.

There are typically many graphs that satisfy one or more of the geometric conditions. Therefore, classes of graphs are regarded instead of individual graphs. A class of graphs contains all graphs that satisfy the same property, here the same geometric condition. The properties are described in form of axioms, which define the graph classes. Transformations and algorithms are applied on the graphs of the so defined graph classes in order to obtain new graphs, which satisfy a certain property. The graphs obtained after applying the transformation form again a graph class. If such a class of graphs satisfies all the axioms of another graph class, then it is contained in the other graph class. A graph class which is contained in another graph class, preserves all the properties of the class it is contained in. In order to prove that a graph class satisfies a certain property, it is therefore sufficient to prove that the graph class is contained in another graph class, which satisfies this property.

The axioms that describe the graph classes can be complex and the methods of checking manually if one class is contained in another one are limited. Therefore, automated reasoning methods are used to decide if a containedness relation between two graph classes holds or not. The automated reasoning methods should provide proofs, in case that the containedness relation holds. In case of non-containedness the automated reasoning methods should return a model of a graph that is contained in one class, but not in the other class.

In order to use automated reasoning methods for the proof tasks and for constructing counterexamples, the properties of the graph classes, the graph transformations and the graph algorithms have to be expressed in a decidable logical fragment, which can be handled by a theorem prover. Since the studied graphs, graph transformations and graph algorithms are based on geometric conditions, underlying geometric properties have to be regarded in addition to the properties of the graphs themselves. In general, studying the use of automated reasoning methods for such geometric graphs is a wide research field. Therefore, the graphs, graph transformations and graph algorithms regarded in this thesis, are limited to properties which are first-order definable. However, already first-order-logic is not decidable in general.

In the following basic concepts of graphs studied in wireless networks and possibilities of using automated reasoning approaches for geometric graphs are introduced. Furthermore, the research questions considered in this thesis are addressed and overview about the structure of the thesis is given.

1.1 Graphs in Wireless Sensor Networks

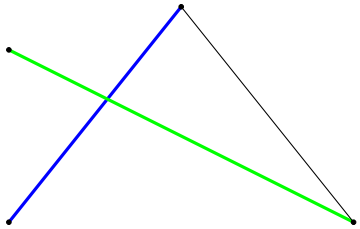
For sending a message in a network a communication path from the sender to the receiver has to exist. If there exists a direct communication link from the sender to the receiver, then the message can be sent directly using this communication link. But what if such a link does not exist? Then the message has to be sent via stopovers to reach the receiver. In this way a path containing multiple links is constructed to send a message from a sender to a receiver. But how is such a path constructed and what conditions are required for the considered network, such that a path exists. This are typical questions considered in the theory of wireless sensor networks (WSNs). In order to analyze a WSN, it can be represented as a graph, where the vertices correspond to the devices and the edges of the graph to the communication links between two devices. For the edges it has to be distinguished if they are directed or undirected. In directed graphs a message can only be sent in one direction, while for undirected graphs the edge can be used to send a message in both directions.

Furthermore, the existence of edges usually depends on the distance between the corresponding vertices. In WSNs, the existence of an edge is more likely for short distances, than for long distances. This can either be obtained by setting strict bounds for the existence resp. non-existence of edges or by using models, where the existence of edges also depends on the existence of other edges or on a random variable.

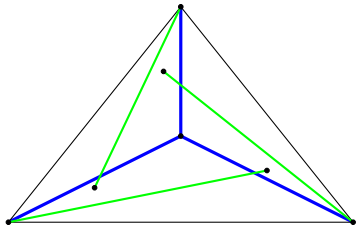
A further important concept for the graphs studied in this context is locality [58]. This means that a vertex can communicate only with other vertices that can be reached by a path up to a certain length. Locality is an important concept for the implementation of distributed algorithms, where every vertex in the graph executes the algorithm based on the information in its neighborhood.

1.2 Application of Plane Graph Drawings

An important property for graphs arising in the context of WSNs is planarity. This property is satisfied if the edges of a graph are not intersecting. For graphs satisfying the planarity condition many routing approaches exist. Such approaches are unicast [15, 42, 44], multicast [28, 65], geocast [70], anycast [54], mobicast [37] and broadcast [67, 71]. An overview about further routing approaches can be found in [51]. Also other approaches in WSNs are based on plane graphs like void and boundary detection [5] , distributed data storage [64, 24], tracking of mobile objects [76], localized address auto-configuration [47], or coordination of mobile sensors [50, 73] . Therefore, constructing a plane subgraph by removing edges from a given graph is an important task, when studying WSNs. Since removing all edges of a given graph, trivially leads to a plane, but disconnected graph, the main challenge is to preserve the connectivity at the same time. This is not possible for all connected graphs, as the following examples in Fig. 1 show. Therefore, the considered graphs need to satisfy some additional structural properties.



(**a**) None of the intersecting edges can be removed without loosing connectivity



(**b**) Removing a green edge or all of the blue edges leads to a disconected graph

Figure 1: Example of graphs, where no plane connected subgraph exists

A simple property for this task is the unit-disk property [17], where all edges exist up to a threshold distance and edges do not exist if the distance between two vertices exceeds this threshold distance. For unit-disk graphs (UDG)s many planarization strategies exist. Some of those planarization strategies are studied in [15, 42, 32, 48]. There exist also some generalizations of UDGs, like quasi unit-disk graphs (QUDG)s [2, 45] and directed transmission graphs (DTG)s [55, 6]. For a QUDG a minimum and maximum communication distance exists, while for a DTG each vertex has an individual communication distance. Other concepts of plane graph drawings are Gabriel graphs [30], relative neighborhood graphs [75] and Delaunay triangulations [25], where edges between two vertices only exits, if certain areas bounded by the corresponding vertices, do not contain other vertices. In this thesis, further possibilities for the construction of connected plane graph drawings are investigated.

Research question 1: Under what conditions, which are likely to apply in wireless networks, is the (local) construction of a connected plane subgraph possible?

1.3 Transformations on and Containedness Between Graph Classes

In order to analyze the properties of graphs occurring in the research of WSNs, classes of graphs can be considered instead of one concrete graph in this class. A class of graphs contains all graphs satisfying a certain property defining the class. For example the class of plane drawings P consists of all graphs without intersecting edges. An important concept for analyzing classes of graphs is to check if a graph class is contained in another graph class and if there exist transformations that transform one class of graphs into another class of graphs. Proving that a graph class A is contained in a graph class B guarantees that the graph class A satisfies all properties of graph class B. In order to investigate whether graph class A satisfies a certain property, it is sometimes simpler to prove that A is contained in a class B, which satisfies the property, than proving directly that A satisfies the property. This concept can for example be used, when the planarity of the graph classes Delaunay triangulation (DT), Gabriel graph (GG) and relative neighborhood graph (RNG) is investigated. In the Euclidean plane with the Euclidean distance a relative neighborhood graph contains a subset of the edges of a Gabriel graph. Furthermore, a Gabriel graph contains a subset of the edges of a

Delaunay triangulation. Since a Delaunay triangulation is always a plane drawing, also a relative neighborhood graph and a Gabriel graph are plane drawings.

For the classes GG and RNG the existence of edges is uniquely defined for a given vertex set, for the class DT, only when no cocircular vertices, i.e. four distinct vertices located on a cycle, exist. Also for a unit disk graph there exists only one graph satisfying the UDG-property for a given vertex set. However, for the class of QUDG there can exist multiple graphs satisfying the QUDG-property for a given vertex set, since for vertices with a distance between the minimal and maximal communication range, the existence of an edge between those vertices is neither forbidden nor enforced. When the same maximum communication range for a UDG and the QUDG is used, then every graph satisfying the UDG property, also satisfies the QUDG-property, but not every graph satisfying the QUDG-property also satisfies the UDG-property. Therefore the class of UDG is contained in the class of QUDG in this case, but not vice versa.

The task for testing the containedness relation between graph classes A and B is therefore to prove that every graph in class A is also contained in class B or to find a so-called witness graph that is contained in A, but not in B. Testing the containedness relation between the graph classes UDG and QUDG was quite simple, since only the minimum and maximum communication radius had to be considered. In general testing the containedness relation between graph classes is a non-trivial task. The reason why this task becomes more complicated, is that the existence of edges in a graph class can also depend on the existence of other edges. One example for such a graph class is the class of plane drawings, where for intersecting segments uv and wx at most one of the edges uv or wx can exist.

Another example is the class of graphs with a maximal clique of size k. For a graph in this class, containing a clique C of size k, no vertex outside the clique can be connected to all vertices of the clique C by an edge. A witness graph that is not contained in the class of graphs with a maximal clique of size k contains therefore $k+1$ vertices. Such a witness graph with $k+1$ vertices is a minimal witness graph in this case, because every subgraph of the witness graph is not a witness graph anymore.

For testing the containedness relation between graph classes a fundamental question is therefore, which size a minimum witness graph can have. In case that the maximal size of a minimum witness graph is n it is enough to consider all graphs up to size n in order to determine if one class is contained in another graph class or not.

The containedness relation between graph classes is not only studied for the graph classes themselves, but also for graph classes obtained after applying a graph transformation γ on a graph class. A graph transformation is not necessarily deterministic and can also be an algorithm for a graph construction. Applying a transformation γ on a graph $A = (V, E)$ yields a graph $B = (W, F)$. The graph class B obtained by applying the transformation γ on the graph class A contains all graphs B, where a graph A in class A exists, such that B is a possible result of applying γ on graph A. An example for a graph transformation is the

construction of a spanning subgraph, where $W = V$ and $F \subseteq E$. Such a transformation is called vertex-preserving and can be used to planarize a given graph, while preserving the connectivity.

In general, there exist also graph transformations, which are not vertex-preserving. Such transformations occur in the construction of overlay graphs. Examples for such overlay graphs are graphs obtained by clustering, edge contraction or vertex contraction. However, the transformations, that are considered in this thesis are usually vertex-preserving. A special case of a vertex-preserving transformation is a simple transformation, where the existence of an edge in the transformed graph only depends on the existence of the corresponding edge and the reversed edge in the graph. In this thesis, testing the containedness relation between graph classes obtained by simple transformations is investigated by using the concept of minimal witness graphs.

Research question 2: Under what conditions can class containedness relationships between graph classes, which are described by certain predicates under simple transformations, be tested by considering only graphs with a fixed maximum number of vertices?

1.4 Using Automated Reasoning Approaches for Geometric Graphs

For analyzing the relation between graph classes and algorithms executed on graphs of a certain graph class, the proof tasks can become quite complicated, because the proofs sometimes require a huge number of case distinctions. Therefore, methods from automated reasoning have to be used to handle these proof tasks. For this purpose the properties of the considered graphs as well as the underlying geometric properties have to be modeled in a way that can be handled by a prover. State of the art approaches mainly consider either automated reasoning in geometry or for graphs. Methods that combine reasoning about geometry and graphs are not known so far. Possibilities to formulate proof tasks for testing the containedness relations for geometric graph classes, in a way that an automated reasoning approach can be used is studied in the following research question.

Research question 3: Can methods from automated reasoning be used to test class containedness relations for graph classes related to planarity conditions and for which distance functions is this possible?

Existing approaches and possibilities for using automated reasoning in geometry and in graph theory are discussed in what follows. For automated reasoning in geometry one can consider conditions using constraints of the coordinates of points [22] or an axiomatization of geometric properties. For the approach using constraints of the coordinates an embedding of the points in a metric space and a weight function for the distance between the points is needed. Such an embedding can be the Euclidean plane with the Euclidean distance as weight function for the length of the edges. However, also other metric spaces or more general weight functions

can be considered. For the axiomatic approach, systems have been proposed by Hilbert [36] and Tarski [74] and were used for theorem proving in geometry [4].

The advantage of such an axiomatic system is that one can formalize geometric conditions, like the condition that two segments intersect or the condition that a point is located inside a triangle as a direct implication by first-order formulae. Another advantage of the axiomatic approach is that, many geometric properties can be described without function symbols and a very restricted form of quantification. Therefore, one often obtains formulae in the Bernays-Schönfinkel class, a fragment of first-order logic without function symbols which is decidable. However, the disadvantage of these approaches is that the obtained models are not intuitive to interpret and some conditions from Euclidean space, e.g. properties for collinear points, do not necessarily hold in an arbitrary metric space [61] proposed by the axioms. Therefore, finding a concrete embedding in the Euclidean space for a model is not always possible or a high computational effort is needed to compute it. The reason why such an embedding might not exist, is that not all properties that are satisfied in the Euclidean space are also implied by the axioms used in the axiomatic approach. A method to compute an embedding of points in the Euclidean plane, for models in an axiom approach is described in [66].

For using automated reasoning in graph theory there exist a lot of studies. In [41] the possibility of describing certain properties of graphs in first-order logic with metric betweenness by Tarski's approach is investigated. Many other of these approaches are based on monadic second-order logic [18, 19, 20, 21] and the verification of graph algorithms uses higher-order logic (HOL) theorem provers [1]. Formalizations of graph properties in Isabelle/HOL [57] can be found in [56, 43]. However, many of the automated reasoning approaches in graph theory consider graphs with a bounded tree-width (e.g. [18, 21]) or graphs with forbidden minors (e.g. [19, 20]). Such properties do not necessarily hold for graphs arising in the study of WSNs. On the other hand, many properties of graphs that are important in the study of WSNs can be defined in first-order logic. However, a property like connectivity is known to be not first-order definable. The reason why connectivity is not first-order definable is that a quantification over paths of The reason why connectivity is not first-order definable is that here a quantification over arbitrary length is required. In this thesis a way of handling connectivity without using higher-order logic is investigated.

Research question 4: How can the properties for geometric graphs be axiomatized, such that the use of higher-order logic can be avoided, when modeling and verifying proof tasks. Especially, is it possible to handle connectivity without using higher-order logic?

A possibility to obtain formulae without second-order quantifiers is second-order quantifier elimination (SOQE), which is an approach for symbol elimination. For SOQE the task is to find for a given formula F_1 an equivalent formula F_2 in first-order logic. When considering graphs obtained after applying transformation it is not always possible to find a first-order

axiomatization of the transformed graph class, even if the transformation and the graph class before applying the transformations is first-order definable. Therefore, a fundamental question is, for which graph classes and which transformations a first-order axiomatization is possible. Such a transformation can also be an algorithm for transforming an input graph from one graph class into an output graph of another graph class. A way to handle proof tasks for the correctness proof of an algorithm by a first-order theorem prover is investigated in the following research question.

Research question 5: Can a first-order theorem prover be used to verify the correctness of a connected planarization algorithm for graphs satisfying certain structural properties?

1.5 Contribution of the Thesis

In this thesis graph classes beyond the UDG class are studied. The possibilities for checking containedness relations between graph classes and proving the correctness of algorithms by using automated reasoning approaches are investigated. In contrast to existing approaches for automated reasoning in graph theory the approaches presented in this thesis avoid to use higher-order logic systems like Isabelle/HOL or Coq. Instead the program H-PILoT[39] for hierarchical proving in local theory extensions, the first-order theorem prover Z3[23] and Mathematica are used to perform the proof task. In order to use those tools, the conditions of the considered graphs and transformations applied on them have to be expressed in first-order logic. The same holds for the conditions of algorithms executed on certain classes of graphs. Therefore, for formulae containing second-order quantifiers, equivalent formulae without those quantifiers have to be derived. It is investigated, which properties of geometric graph classes can be expressed in first-order logic. Of special interest is the connectivity of graphs, since this property is known to be not first-order definable. In order to express the connectivity of graphs a pointer structure with linked lists is used to reason about certain paths that guarantee the connectivity of the graph.

The following properties of graph classes, transformations and algorithms are considered in this thesis:

- Graph class containedness relations between Gabriel graphs, relative neighborhood graphs and plane drawings for several distance functions
- Graph class containedness relations of graph classes defined by axioms for the existence and non-existence of edges under simple transformations
- Algorithms for the connected planarization of graphs in graph classes beyond UDG.

Graph class containedness relations between Gabriel graphs(GG), relative neighborhood graphs(RNG) and plane drawings(P) for several distance functions

Based on a method from [13], which uses H-PILoT for testing class containedness relations between Gabriel graphs, relative neighborhood graphs and plane drawings for the Euclidean distance and an arbitrary metric, the possibilities of applying this method for further distance functions is investigated. H-PILoT is used to construct counterexamples, which show that a relative neighborhood graph is not guaranteed to be a plane drawing anymore, when the distance functions Manhattan and maximum metric are used instead of the Euclidean distance.

Afterwards the class containedness relations are investigated for general distance functions induced by a (quasi-)norm. For those general distance functions H-PILoT cannot be used anymore and the proofs have to be performed manually. It is proven that for the remaining cases of metrics induced by a norm $\mathsf{RNG} \subseteq \mathsf{GG}$ and therefore also $\mathsf{RNG} \subseteq \mathsf{GG} \subseteq \mathsf{P}$ holds, since $\mathsf{GG} \subseteq \mathsf{P}$ holds for every metric. For distance functions induced by a quasi norm with p between 0 and 1 it is proven that, $\mathsf{RNG} \not\subseteq \mathsf{GG}$ and also $\mathsf{GG} \not\subseteq \mathsf{P}$ as well as $\mathsf{RNG} \not\subseteq \mathsf{P}$. For an arbitrary quasi-norm with a relaxed version of the triangle inequality the method from [13] is used again to construct two models showing that $\mathsf{GG} \not\subseteq \mathsf{P}$ and $\mathsf{RNG} \not\subseteq \mathsf{P}$.

Graph class containedness relations of graph classes defined by axioms for the existence and non-existence of edges under simple transformations

The possibilities of deriving first-order formulae for testing the containedness relation between transformed graph classes are investigated. The considered graph classes are defined by inclusion axioms, for the existence of edges, exclusion axioms, for the non-existence of edges and transfer axioms, for the existence of edges based on the existence of other edges. The first-order formulae for testing the class containedness relations contain only the predicates defining the axioms of a source graph class and edges of the transformed graph class, while the predicates for edges of the source graph are eliminated. If possible, such formulae are derived manually and conditions for testing the containedness relation between graph classes are obtained by those formulae. The transformations considered in this context are simple transformations, where the existence of an edge in the transformed graph depends only on the existence of the corresponding edge and the reversed edge in the source graph. In total there exists 16 simple vertex-preserving transformations.

It is shown that in general only for 10 of 16 simple transformations a counterexample with at most 4 vertices is sufficient to test whether a graph is contained in this transformed graph or not. For those 10 graph classes conditions for the constructions of counterexamples are derived. For the remaining transformations counterexamples, showing that the maximal size of a minimum witness graph can be arbitrary large, are constructed.

Then the concept is first used for manually testing the containedness relation between certain graph classes for different distance functions. Afterwards the derived formulae are used to check the containedness relation between all possible graph classes based on inclusion, exclusion and transfer predicate for simple transformations, when the Euclidean distance is used. For this Mathematica is used to either prove that a class is contained in another class or construct a counterexample.

Planarization of graphs while preserving connectivity
The class of redundancy and coexistence graphs(RCG) is a class of graphs beyond UDG, defined by the properties redundancy and coexistence. The redundancy property is satisfied if for each pair of intersecting edges, at least one of the four corresponding vertices forming the intersection, is connected by an edge to the other three vertices. The coexistence property is satisfied if each vertex located inside a triangle formed by three edges is connected by an edge to the border vertices of the triangle. For the class RCG it is first proven manually that a connected and plane spanning subgraph exists by presenting a recursive approach to construct such a spanning subgraph.

In a next step an axiom system for modeling parts of Euclidean geometry, which are required to prove the correctness of a local algorithm for the connected planarization of graphs in the class RCG is defined. This axiom system is related to an axiom system of Hilbert and it is proven that some of Hilbert's axioms, like the axiom of Pasch are also satisfied in this axiom system. The axiom system is then extended by axioms for expressing properties of vertices and edges of geometric graphs. In a further step the axiom systems is extended by axioms that express properties for a graph constructed by a planarization algorithms and properties expressing conditions required for the correctness proof of this algorithm.

The proof steps of the correctness proof of the algorithm are then verified by the theorem prover Z3. In order to prove that the considered graph properties redundancy and coexistence cannot be simplified significantly models for counterexamples are constructed by Z3, when the properties are weakened. Furthermore, Z3 is used to show that also the proof tasks cannot be simplified significantly, by constructing models for counterexamples when the assumptions in the proof task are weakened by reducing the number of case distinctions considered in the proof.

Overview of Contributions The contributions of the thesis are stated in the following. Each of the contributions corresponds to one research question

Contribution1: It is proven that there exists a connected and plane subgraph for graphs satisfying the acyclic redundancy, a special version of the redundancy property, and also for graphs satisfying redundancy and coexistence. Furthermore, it is proven that for the construction of such subgraphs local algorithms exist.

Contribution2: Expressions for testing the class containedness relations for graph classes obtained under simple transformations by considering only graphs up to 4 vertices are derived, in the case that the predicates are defined based on the Euclidean distance.

Contribution3: A method from hierarchical reasoning is used to test class containedness relations for the Gabriel graph, relative neighborhood graph and plane drawings, for the Euclidean distance and further distance functions obtained from a p-(quasi)norm

Contribution4: An axiom system for geometric graphs similar to Hilbert's axiomatization of geometry is proposed. In this axiom system connectivity is handled by using linked lists to define certain paths that guarantee the connectivity of a graph.

Contribution5: The first-order theorem prover Z3 is used to verify the proof steps in a manually derived correctness proof of a connected planarization algorithm for the class of RCG.

1.6 Publications

This thesis is based on the following publications:

- Lucas Böltz and Hannes Frey. Existence of Connected Intersection-Free Subgraphs in Graphs with Redundancy and Coexistence Property. In 15th International Symposium on Algorithms and Experiments for Wireless Sensor Networks (ALGOSENSORS), volume 11931 LNCS, pages 63–78, 2019.

- Steffen Böhmer, Lucas Böltz, and Hannes Frey. Local Construction of Connected and Plane Spanning Subgraphs under Acyclic Redundancy. In 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOPT 2020

- Lucas Böltz, Benjamin Becker, and Hannes Frey. Local Construction of Connected Plane Subgraphs in Graphs Satisfying Redundancy and Coexistence. In XI Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS), pages 1–10, 2021.

- Lucas Böltz, Hannes Frey, Dennis Peuter, and Viorica Sofronie-Stokkermans. On testing containedness between geometric graph classes using second-order quantifier elimination and hierarchical reasoning (short paper). In Proceedings of the Second Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2021) associated with the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021), Online Event, November 4, 2021, volume 3009 of CEUR Workshop Proceedings, pages 37–45. CEUR-WS.org, 2021.

- Steffen Böhmer, Lucas Böltz, and Hannes Frey. WIP: local heuristics for very likely connected and intersection free wireless network topologies under log-normal shadowing. In 23rd IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2022, Belfast, United Kingdom, June 14-17, 2022, pages 165–168. IEEE, 2022.

- Lucas Böltz and Hannes Frey. Automatically testing containedness between geometric graph classes defined by inclusion, exclusion, and transfer axioms under simple transformations. Inf., 13(12):578, 2022.

- Lucas Böltz and Hannes Frey. Automatically testing containedness between geometric graph classes defined by inclusion, exclusion, and transfer axioms under simple transformations. CCCG 2021.

- Lucas Böltz, Viorica Sofronie-Stokkermans, and Hannes Frey. On the verification of the correctness of a subgraph construction algorithm. In Rayna Dimitrova, Ori Lahav, and Sebastian Wolff, editors, 25th International Conference on Verification, Model Checking, and Abstract Interpretation, Lecture Notes in Computer Science. Springer, 2024.

Table 1 also lists these papers in chronological order and denotes which chapter of the thesis the contents of the papers is related to.

	Title	Authors	Published in	Related to	Ref.
1	Existence of Connected Intersection-Free Subgraphs in Graphs with Redundancy and Coexistence Property	Lucas Böltz, Hannes Frey,	ALGO 2019	Chapter 5	[11]
2	Local Construction of Connected and Plane Spanning Subgraphs under Acyclic Redundancy	Steffen Böhmer, Lucas Böltz, Hannes Frey	WiOpt 2020	Chapter 5,	[7]
3	Local Construction of Connected Plane Spanning Subgraphs in Graphs Satisfying Redundancy and Coexistence	Lucas Böltz, Benjamin Becker, Hannes Frey	LAGOS 2021	Chapter 5 Chapter 7	[10]
4	On Testing Containedness Between Geometric Graph Classes using Second-order Quantifier Elimination and Hierarchical Reasoning (Short Paper)	Lucas Böltz, Hannes Frey, Dennis Peuter, Viorica Sofronie-Stokkermans	SOQE 2021	Chapter 3	[13]
5	Automatically testing containedness between geometric graph classes defined by inclusion, exclusion, and transfer axioms under simple transformations	Lucas Böltz, Hannes Frey,	CCCG 2021, Information 2022	Chapter 4	[29] [12]
6	Local heuristics for very likely connected and intersection free wireless network topologies under log-normal shadowing (Work in Progress)	Steffen Böhmer, Lucas Böltz, Hannes Frey	WoWMoM 2022	Chapter 5	[8]
7	On the verification of the correctness of a subgraph construction algorithm	Lucas Böltz, Viorica Sofronie-Stokkermans, Hannes Frey	VMCAI 2024	Chapter 6 Chapter 7	[14]

Table 1: List of publications

1.7 Structure of the Thesis

In **Section 2** some basic graph concepts occurring in wireless network research and the syntax of first-order logic and theories used for automated reasoning are introduced.

In **Section 3** the graph classes RNG and GG and the class of plane drawings are considered for arbitrary distance functions. Starting with the Euclidean distance a method from [13] is presented, which proves the relation $\mathsf{RNG} \subseteq \mathsf{GG} \subseteq \mathsf{P}$ using H-PILoT. Furthermore, it was proven with this method for an arbitrary metric that, $\mathsf{RNG} \not\subseteq \mathsf{GG}$, while $\mathsf{GG} \subseteq \mathsf{P}$ still holds. Based on the method from [13] the containedness relations between RNG, GG and P are studied for further distance functions. If possible, these tasks are performed using H-PILoT and if H-PILoT cannot be used the proofs and counterexamples are derived manually.

In **Section 4** the relation of graph classes described by inclusion, exclusion and transfer axioms under simple transformations are studied. First-order formulae for testing the containedness relation between the transformed graph classes are derived by manually applying symbol elimination. The derived formulae are then used to check the containedness relation between all possible graph classes based on inclusion, exclusion and transfer predicate for simple transformations, when the Euclidean distance is used. For this Mathematica is used to either prove that a class is contained in another class or construct a counterexample.

In **Section 5** A local algorithm for the construction of a connected and plane subgraph for graphs satisfying a special variant of the redundancy property, the acyclic redundancy is presented. For graphs in the class RCG, first the existence of a connected and plane spanning subgraph is proven. Afterwards the CP-algorithm [3] for constructing such a connected and plane spanning subgraph based on local information is presented.

In **Section 6** a suitable model of the underlying geometry, the graph properties and the properties used in the correctness proof of the CP-algorithm is introduced. The geometric axioms of the proposed axiom system are related to Hilbert's axiom system and it is proven that some of Hilbert's axioms also hold in the proposed axiom system.

In **Section 7** the proof steps of a manually derived correctness proof of the CP-algorithm are verified, by using a suitable encoding of the problems and satisfiability checks, which are solved by the theorem prover Z3. In order to analyze whether the proof can still be carried for relaxed graph properties or whether the proof tasks can be simplified by reducing the number of case distinctions, models for counterexamples are constructed by Z3 for a weaker version of the coexistence property and simplification of certain proof steps.

Finally an overview of the results of this thesis is given in **Section 8**. The possibilities of using automated reasoning for other graph classes and graph algorithms in future work are discussed here.

In **Appendix 9** additional tables, figures and full details of proofs can be found, which were omitted in the previous sections of the thesis.

2 Preliminaries

This thesis covers the research areas of automated reasoning, graph theory and computational geometry. Therefore, an introduction of notations used in each of the areas is required. First basic concepts of first-order logic are introduced. Afterwards theories and theory extensions used in automated reasoning approaches are defined. After defining the notations used for automated reasoning, the graphs and the transformation, which are investigated in this thesis are introduced. Since the graphs and graph transformations considered in this thesis are geometric graphs, also the underlying geometric concepts have to be considered.

2.1 First-order Logic

Standard notation from first-order-logic is assumed to be known. Further information can be found in [27].

A *signature* $\Pi = (\Sigma, \mathsf{Pred})$, consists of a set Σ of *function symbols* and a set Pred of *predicate symbols*, where for each function symbol and predicate symbol the arity is specified. A function symbol with arity zero is called a *constant* and a predicate symbol with arity zero is called a *propositional variable*. The set X of *variables* is countably infinite.

The set of Π-*terms* $T_\Pi(X)$ over X is defined as the smallest set containing

- all variables in X;
- all constants in Π;
- $f(t_1, ..., t_n)$ if f is a function in Π with arity n and every t_i is a Π-term for $i \in \{1, ..., n\}$.

A term is called a *ground term* if it does not contain variables. The set of ground terms is denoted by T_Π.

A Π-*atom* is an equality $t_1 \approx t_2$ between terms or an expression $P(t_1, ..., t_n)$, where P is a predicate symbol of arity n and t_i is a Π-term for every $i \in \{1, ..., n\}$. A Π-*literal* is a Π-atom A (positive literal) or a negated Π-atom $\neg A$ (negative literal).

The set of Π-*formulae* $F_\Pi(X)$ with variables in X is defined as the smallest set containing

- $\top$, $\bot$, and all Π-atoms;
- $\neg F$ if F is a Π-formula;
- $F_1 \circ F_2$ if F_1 and F_2 are Π-formulae and $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$;
- $\forall x F$ and $\exists x F$ if F is a Π-formula and $x \in X$.

A formula not containing any quantifiers is called *quantifier-free*. A formula not containing any variables is called a *ground formula*. The set of ground formulae is denoted with F_Π.

A Π-*clause* is a disjunction of Π-literals. A clause is called a *ground clause* if it does not contain variables. A *Horn-clause* is a clause which contains at most one positive literal.

A formula is in *conjunctive normal form* (CNF) if it is a conjunction of clauses (i.e. if it is a conjunction of disjunctions of literals); it is in *disjunctive normal form* (DNF) if it is a disjunction of conjunctions of literals.

In QxF with $Q \in \{\exists, \forall\}$, F is called the *scope* of the quantifier Qx. A variable x is called a *bound variable* of a formula F if it appears in the scope of a quantifier Qx in F. A variable that is not bound by a quantifier is called a *free variable* of F. A Π-formula is called a Π-*sentence* if every variable is bound by a quantifier. It is in *prenex normal form* if it consists of a string of quantifiers and bound variables followed by a quantifier-free formula. It is called a *universal formula* if it is in prenex normal form and contains only universal quantifiers.

Semantics

A Π-*structure* for a signature $\Pi = (\Sigma, \mathsf{Pred})$ is a tuple $\mathcal{A} = (|\mathcal{A}|, \Sigma^{\mathcal{A}}, \mathsf{Pred}^{\mathcal{A}})$, where $|\mathcal{A}|$ is the universe, $\Sigma^{\mathcal{A}}$ the interpretation of the set of functions and $\mathsf{Pred}^{\mathcal{A}}$ interpretation of the set of predicates. A *valuation* over a Π-structure $\mathcal{A}$ is a total map $\beta : X \to |\mathcal{A}|$, assigning to each variable an element of the universe. $\beta[x \to a]$ is denoted as the valuation that maps x to a and every other variable $y \in X$ to $\beta(y)$. The value $\mathcal{A}(\beta)(t)$ of a term t in $\mathcal{A}$ w.r.t. β is defined inductively by:

- $\mathcal{A}(\beta)(t) = \beta(t)$ if $t \in X$,
- $\mathcal{A}(\beta)(t) = c_{\mathcal{A}}$ if t is a constant c,
- $\mathcal{A}(\beta)(t) = f_{\mathcal{A}}(\mathcal{A}(\beta)(t_1), ..., \mathcal{A}(\beta)(t_n))$ if $t = f(t_1, ..., t_n)$.

The set of truth values is $\{0, 1\}$, where 0 stands for *false* and 1 for *true*. If $\mathcal{A}$ is a Π-structure and β a valuation, then $\mathcal{A}(\beta)(F)$, the truth value of a formula F in $\mathcal{A}$ w.r.t. β, is defined inductively over the structure of F by

- $\mathcal{A}(\beta)(\top) = 1$,
- $\mathcal{A}(\beta)(\bot) = 0$,
- $\mathcal{A}(\beta)(P(t_1, ..., t_n)) = 1$ if and only if $(\mathcal{A}(\beta)(t_1), ..., \mathcal{A}(\beta)(t_n)) \in P_{\mathcal{A}}$,
- $\mathcal{A}(\beta)(\neg F) = 1 - \mathcal{A}(\beta)(F)$,
- $\mathcal{A}(\beta)(F \wedge G) = \min(\mathcal{A}(\beta)(F), \mathcal{A}(\beta)(G))$,
- $\mathcal{A}(\beta)(F \vee G) = \max(\mathcal{A}(\beta)(F), \mathcal{A}(\beta)(G))$,

- $\mathcal{A}(\beta)(F \to G) = 0$ if and only if $\mathcal{A}(\beta)(F) = 1$ and $\mathcal{A}(\beta)(G) = 0$,
- $\mathcal{A}(\beta)(\exists x F) = \max_{a \in |\mathcal{A}|}(\mathcal{A}(\beta[x \to a])(F))$,
- $\mathcal{A}(\beta)(\forall x F) = \min_{a \in |\mathcal{A}|}(\mathcal{A}(\beta[x \to a])(F))$.

A structure $\mathcal{A}$ is called a *model* of a formula F, denoted $\mathcal{A} \models F$, if and only if $\mathcal{A}(\beta)(F) = 1$ for every valuation $\beta : X \to |\mathcal{A}|$. F is called *valid*, denoted $\models F$, if and only if $\mathcal{A} \models F$ for every structure $\mathcal{A}$. F is called *satisfiable* if and only if there exist $\mathcal{A}$ and β such that $\mathcal{A}(\beta)(F) = 1$, otherwise it is called *unsatisfiable*. If F and G are formulae, then F entails G, denoted $F \models G$, if and only if for all structures $\mathcal{A}$ and all valuations β it holds that $\mathcal{A}(\beta)(F) = 1$ implies $\mathcal{A}(\beta)(G) = 1$. $F \models \bot$ means that F is unsatisfiable. A set of formulae N is satisfiable if and only if there exist $\mathcal{A}$ and β such that $\mathcal{A}(\beta)(F) = 1$ for all $F \in N$, otherwise it is called unsatisfiable. A set of formulae N entails a formula G, denoted $N \models G$ if $\mathcal{A}(\beta)(F) = 1$ for all $\mathcal{A}, \beta$ and for all $F \in N$ implies $\mathcal{A}(\beta)(G) = 1$. Two formulae F and G are called *equivalent*, denoted as $F \equiv G$, if $F \models G$ and $G \models F$.

Unsatisfiable Cores If a set of formulae N is unsatisfiable, often not every formula in N is responsible for the unsatisfiability of N. To make reasoning more efficient it can therefore be of advantage to compute a subset of N that is still unsatisfiable. Such an unsatisfiable subset of N is called an unsatisfiable core of N.

Definition 1 (**(Minimal) Unsatisfiable Core).** *An unsatisfiable core of an unsatisfiable set of formulae N is a subset $U \subseteq N$ such that U is also unsatisfiable. An unsatisfiable core U is called minimal if every proper subset $V \subset U$ is satisfiable.*

2.2 Theories and Local Theory Extensions

A logical theory can be seen as a collection of sentences, defining which formulae are supposed to be true in the theory. This can for instance be done by defining a set of axioms and considering all logical consequences of the axioms or by giving a set of models in which the statements of the theory hold. The formal definitions are given below.

A Π-*theory* $\mathcal{T}$ is a set of Π-sentences. A set $Ax_{\mathcal{T}}$ of Π-sentences is an *axiomatization* of a Π-theory $\mathcal{T}$ if and only if $\mathcal{T}$ and $Ax_{\mathcal{T}}$ have the same consequences, i.e. if for every Π-sentence F it holds that $\mathcal{T} \models F$ if and only if $Ax_{\mathcal{T}} \models F$. The elements of $Ax_{\mathcal{T}}$ are called *axioms*.

If F and G are formulae $F \models_{\mathcal{T}} G$ (also written as $\mathcal{T} \cup F \models G$) is used to express that every model of F which is also a model of $\mathcal{T}$ is a model of G. $F \models_{\mathcal{T}} \bot$ means that there is no model of $\mathcal{T}$ in which F is true. If there is a model of $\mathcal{T}$ which is also a model of F, then F is called $\mathcal{T}$-*consistent* (or satisfiable w.r.t. $\mathcal{T}$). If $F \models_{\mathcal{T}} G$ and $G \models_{\mathcal{T}} F$, then F and G are called equivalent w.r.t. $\mathcal{T}$, denoted by $F \equiv_{\mathcal{T}} G$.

A theory $\mathcal{T}$ is called *locally finite* if for every finite set of variables X the set of terms $T_\Pi(X)$ contains only finitely many terms $t_1, ..., t_n$ with $\models_\mathcal{T} t_i \not\approx t_j$ for $i \neq j$.

Often combinations or extensions of theories have to be considered. Combining two theories $\mathcal{T}_1$ and $\mathcal{T}_2$ means that the corresponding axioms of the theories are combined. Sometimes a base theory is extended with additional function symbols which have certain properties (decribed by universal formulae). This is then called a theory extension. Of special interest are theory extensions that have a "locality" property. Such local theory extensions and possibilities of hierarchical reasoning in local theory extensions is described in the following.

2.2.1 Local Theory Extensions

The notion of *locality* was first introduced by Givan and McAllester in [33, 34], who looked into so-called *local inference relations* for which checking validity of ground Horn clauses can be done in polynomial time. Ganzinger continued this work in [31] by studying *local theories* and established links between embeddability properties and locality. In [69] Sofronie-Stokkermans further generalized these results by introducing the notion of *local theory extensions*. In [38] Sofronie-Stokkermans et al. introduced the notion of Ψ*-local theory extensions*, where Ψ is a closure operator on ground terms, which were further studied by Ihlemann and Sofronie-Stokkermans in [40]. In the following the notions for a theory extension, local theory extension and Ψ-local theory extension are defined.

Definition 2 (Base Theory and Theory Extension). *Let* $\Pi_0{=}(\Sigma_0, \mathit{Pred})$ *be a signature and* $\mathcal{T}_0$ *be a theory with signature* Π_0 *(*Σ_0 *are called* base functions*). A theory extension* $\mathcal{T}_1 := \mathcal{T}_0 \cup \mathcal{K}$ *of a base theory* $\mathcal{T}_0$ *is a theory with new function symbols in a set of extension functions* Σ *disjoint from the base functions* Σ_0*. The properties of the extension functions are axiomatized using a set* $\mathcal{K}$ *of clauses in the extended signature* $\Pi_1 = (\Sigma_0 \cup \Sigma, \mathit{Pred})$*, which contain function symbols in* Σ*.*

If G is a finite set of ground Π_1^C-clauses (where $\Pi_1^C = (\Sigma_0 \cup \Sigma \cup \Sigma_C, \mathsf{Pred})$ is the extension of Π_1 with constants in a countable set Σ_C of fresh constants) and $\mathcal{K}$ a set of Π-clauses, $\mathsf{st}(\mathcal{K}, G)$ is denoted as the set of all ground terms which occur in G or $\mathcal{K}$, and $\mathsf{est}(\mathcal{K}, G)$ the set of all extension ground terms (i.e. ground terms whose root is in Σ) which occur in G or $\mathcal{K}$. Every finite set G of ground clauses is regarded as the ground formula $\bigwedge_{C \in G} C$ here.

The set of instances of $\mathcal{K}$, in which all terms starting with an extension function are ground terms in the set $\mathsf{est}(\mathcal{K}, G)$ is denoted by $\mathcal{K}[G]$, i.e.

$$\begin{aligned}\mathcal{K}[G] = \{C\sigma \mid\ & C \in \mathcal{K} \text{ and } \sigma : X \to X \cup T_{\Sigma \cup \Sigma_C} \text{ is such that} \\ & \text{(i)}\ \text{ for each extension term } f(t) \text{ in } C, f(t)\sigma \in \mathsf{est}(\mathcal{K}, G) \\ & \text{(ii)}\ x\sigma = x \text{ if } x \text{ does not occur in an extension term}\}.\end{aligned}$$

Definition 3 (**Local Theory Extension).** *A theory extension $\mathcal{T}_0 \cup \mathcal{K}$ is* local *if the condition* ($\mathsf{Loc_f}$) *is satisfied:*

($\mathsf{Loc_f}$) *For every finite set G of ground clauses in the signature Π_1^C it holds that $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \bot$ if and only if $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$ is unsatisfiable.*

It is always true that $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G \models \bot$ implies $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \bot$. However, in general the satisfiability of $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$ does not imply the satisfiability of $\mathcal{T}_0 \cup \mathcal{K} \cup G$. If this is the case, i.e. restricting to a "local" part of the problem for checking (un-)satisfiability, then the theory extension has the locality property. Sometimes the instances $\mathcal{K}[G]$ computed from the set of ground terms $\mathsf{est}(\mathcal{K}, G)$ may not be sufficient for guaranteeing satisfiability. In this case one can try to find a suitable closure operator Ψ to compute a larger set of ground terms. If this guarantees satisfiability, then this property is called Ψ-locality.

Definition 4 (**Term Closure Operator [40]).** *Let Ψ be a map associating with $\mathcal{K}$ and a set of Π^C-ground terms T a set $\Psi_\mathcal{K}(T)$ of Π^C-ground terms. $\Psi_\mathcal{K}$ is called a* term closure operator *if the following holds for all sets of ground terms T and T':*

1. $\mathsf{est}(\mathcal{K}, T) \subseteq \Psi_\mathcal{K}(T)$,
2. *If $T \subseteq T'$, then $\Psi_\mathcal{K}(T) \subseteq \Psi_\mathcal{K}(T')$,*
3. $\Psi_\mathcal{K}(\Psi_\mathcal{K}(T)) = \Psi_\mathcal{K}(T)$,
4. *for any map $h : C \to C$ it holds that $\overline{h}(\Psi_\mathcal{K}(T)) = \Psi_\mathcal{K}(\overline{h}(T))$, where $\overline{h}$ is the canonical extension of h to extension ground terms.*

The important case, in which this criterion is helpful, is the case in which for every finite set T of ground terms the set $\Psi_\mathcal{K}(T)$ is also finite. This is considered in the following. Let the set $\Psi_\mathcal{K}(\mathsf{est}(\mathcal{K}, G))$ of ground terms derived using the closure operator $\Psi_\mathcal{K}$ be denoted by $\Psi_\mathcal{K}(G)$.

Definition 5 (Ψ**-local Theory Extension [38, 40]).** *A theory extension $\mathcal{T}_0 \cup \mathcal{K}$ is* Ψ-local *if the condition* ($\mathsf{Loc_f^\Psi}$) *is satisfied:*

($\mathsf{Loc_f^\Psi}$) *For every finite set G of ground clauses in the signature Π_1^C it holds that $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \bot$ if and only if $\mathcal{T}_0 \cup \mathcal{K}[\Psi_\mathcal{K}(G)] \cup G$ is unsatisfiable.*

2.2.2 Hierarchical Reasoning in Local Theory Extensions

For (Ψ)-local theory extensions hierarchical reasoning is possible. Only the case of local theory extensions is discussed here, but similar results hold also for Ψ-local theory extensions under the additional assumption that for every finite set G of ground clauses $\Psi_\mathcal{K}(G)$ is finite. If $\mathcal{T}_0 \cup \mathcal{K}$ is a local extension of $\mathcal{T}_0$ and G is a set of ground $\Sigma_0 \cup \Sigma_1 \cup \Sigma_C$-clauses, then $\mathcal{T}_0 \cup \mathcal{K} \cup G$ is unsatisfiable if and only if $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$ is unsatisfiable. This last satisfiability test can be reduced to a satisfiability test w.r.t. $\mathcal{T}_0$.

The idea is to purify $\mathcal{K}[G] \cup G$ by applying the following steps:

(i) In a bottom-up manner new constants c_t for subterms $t = f(g_1, \ldots, g_n)$ are introduced, where $f \in \Sigma$ and g_i are ground $\Sigma_0 \cup \Sigma_C$-terms, and replace the terms t with the constants c_t.

(ii) Definitions $c_t \approx t$ are added to a set Def.

By $\mathcal{K}_0 \cup G_0 \cup \mathsf{Def}$ the set of formulae obtained this way is denoted. Then G is satisfiable w.r.t. $\mathcal{T}_0 \cup \mathcal{K}$ if and only if $\mathcal{K}_0 \cup G_0 \cup \mathsf{Con}_0$ is satisfiable w.r.t. $\mathcal{T}_0$, where Con_0 is the following set of instances of congruence axioms:

$$\mathsf{Con}_0 = \{(\bigwedge_{i=1}^{n} c_i \approx d_i) \rightarrow c \approx d \mid c \approx f(c_1, \ldots, c_n), d \approx f(d_1, \ldots, d_n) \in \mathsf{Def}\}.$$

Theorem 1 ([69]). *If $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ is a local theory extension of $\mathcal{T}_0$ and G is a finite set of ground clauses, then the problem of checking whether G is satisfiable w.r.t. $\mathcal{T}_0 \cup \mathcal{K}$ can be reduced to checking the satisfiability w.r.t. $\mathcal{T}_0$ of the set of formulae $\mathcal{K}_0 \cup G_0 \cup \mathsf{Con}_0$ constructed as explained above. If $\mathcal{K}_0 \cup G_0 \cup \mathsf{Con}_0$ belongs to a decidable fragment of $\mathcal{T}_0$, the decision procedure for this fragment can be used to decide whether $\mathcal{T}_0 \cup \mathcal{K} \cup G$ is unsatisfiable.*

As the size of $\mathcal{K}_0 \cup G_0 \cup \mathsf{Con}_0$ is polynomial in the size of G (for a given $\mathcal{K}$), locality allows to express the complexity of the ground satisfiability problem w.r.t. $\mathcal{T}_1$ as a function of the complexity of the satisfiability of formulae w.r.t. $\mathcal{T}_0$.

2.3 Graph Models and Algorithms in Wireless Networks

In the following graphs are regarded as ordered pairs $G = (V, E)$, where V represents a finite set of vertices and E the set of edges connecting two vertices. The set of edges is either a set of ordered or a set of unordered pairs of vertices. In an *undirected graph* an edge corresponds to an unordered pair of two vertices, i.e $E \subseteq \{\{u, v\} | u, v \in V\}$. On the other hand, in a *directed graph* an edge corresponds to an ordered pair of two vertices, i.e $E \subseteq \{(u, v) | (u, v) \in V^2\}$.

All considered graphs are assumed to have no self-loops, i.e. for all $u \in V, uu \notin E$. A graph $G = (V, E)$, in which an edge exists between each pair of distinct vertices is a *complete graph* if G is undirected resp. a *complete directed graph* if G is directed. A complete graph with n vertices is denoted as K_n. The *complement of a graph* $G = (V, E)$ is a graph $\overline{G} = (V, \overline{E})$, where for each pair of distinct vertices, an edge in the complement graph exists if and only if the edge does not exist in G, i.e for all $u, v \in V : uv \in \overline{E} \leftrightarrow u \neq v \wedge uv \notin E$. Therefore, $E \cap \overline{E} = \emptyset$ and $E \cup \overline{E}$ corresponds to the edge set of a complete graph.

The vertices of the graphs are elements in a set X, which is typically the set of points in $\mathbb{R}^2$. In case that X is $\mathbb{R}^2$ the graph G is termed a *embedded graph* in the Euclidean plane. Sometimes X also refers to a set of points in a space (X, d) with a weight (usually a distance) function d. A *weighted graph* (G, d) is a graph G assigned with a weight (typically a distance)

function. In order to distinguish between a general weight function and a distance function, c is used as function symbol for weight functions and d is used as function symbol for distance functions. The relations $<, \leq, >, \geq, =$ are used to compare distance resp. weights. A weighted graph (G, c) assigned with an injective weight function c, i.e. $c(u, v) \neq c(w, x)$ if $u \neq w$ or $v \neq x$, is denoted as a *injective weighted graph*. If not further specified X is $\mathbb{R}^2$ and d refers to the Euclidean distance. In this case (G, d) is termed a *weighted embedded graph*.

For modeling graphs in first-order logic, a unary predicate V for the vertices and a binary predicate E for the edges is used, such that $V(x)$ is true iff $x \in V$ and $E(x, y)$ is true iff $xy \in E$. A special case of a directed graph is a *symmetric graph*, where either none or both of the edges uv and vu exist, i.e. for all $u, v \in V : E(u, v) \leftrightarrow E(v, u)$. The construction of a symmetric graph is used, when a directed graph is transformed into an undirected graph. On the other hand transforming an undirected graph into a directed graph, by assigning each edge with an orientation results in a *oriented graph*. For an oriented graph at most one of the edges uv and vu can exist, i.e. for all $u, v \in V : E(u, v) \rightarrow \neg E(v, u)$.

For directed and undirected graphs, one sometimes investigates, whether a graph contains a certain pattern, where some edges exist, while other edges do not exist and for all remaining vertex pairs the existence of an edge is not specified. For this a *graph pattern* is defined by $P(V, S, T)$, where S and T are disjoint subsets of ordered resp. unordered pairs of distinct vertices in V^2. A graph pattern $P(V, S, T)$ is contained in a graph $G = (V, E)$ iff $S \subseteq E$ and $T \subseteq \overline{E}$. The size of a graph pattern is defined as $|S| + |T|$.

In wireless network theory often classes of graphs satisfying certain properties are studied. Such a graph class C contains all graphs $G = (V, E)$ that satisfy a property P defining the graph class, e.g. a geometric condition. For a graph $G = (V, E)$ which is contained in a class C the notation $G \in \mathsf{C}$ is used. A graph class A is contained in a graph class B iff all graphs from class A are also in class B, i.e. $\mathsf{A} \subseteq \mathsf{B}$ iff for all $G : (G \in \mathsf{A}) \rightarrow (G \in \mathsf{B})$

Graphs and classes of graphs can also be considered after applying transformations on them. In general a graph transformation γ transforms a source graph $G = (V, E)$ into an image graph $H = (W, F)$. Such a graph transformation can also be an algorithm, for the construction of a graph with certain properties. In this case γ is not necessarily a deterministic transformation. When applying a transformation γ on a class of graphs C this results in a new graph class $\gamma(\mathsf{C}) = \{(W, F) \mid \exists G = (V, E) \in \mathsf{C} : (W, F) = \gamma(G)\}$.

The axiomatization of a graph class $\gamma(\mathsf{C})$ obtained by applying a transformation is based on the axiomatization of the graph class C and the axiomatization of the transformation γ. Let C be a graph class axiomatized by axioms $\mathsf{Ax}_{\mathsf{C}}(V, E)$ and let γ be a transformation of graph classes, described by a set $\mathsf{T}_{\gamma}(V, E, W, F)$ of formulae. To find a first-order axiomatization of $\gamma(\mathsf{C})$, which does not refer to C, i.e. the axiomatization does not contain the predicate symbols V and E, a first-order formula equivalent to $\exists V \exists E(\mathsf{Ax}_{\mathsf{C}}(V, E) \wedge \mathsf{T}_{\gamma}(V, E, W, F))$ is required. This is a second-order quantifier elimination problem, which is a non-trivial task. An approach for using second-order quantifier elimination for graph transformations is investigated in [61].

After defining the considered graphs and graph transformations in general, some specific graphs, graph classes and graph transformations studied in the context of wireless networks are defined in the following.

2.3.1 Basic Graph Classes in Wireless Network Models

First, an overview about some basic graph models considered in the research of wireless network theory is presented. Initial theoretical studies modeled such networks as *unit disk graphs* (UDG), where network vertices are connected iff their Euclidean distance is less or equal than 1 and vertices are not connected by self-loops.

Definition 6 (**Unit Disk Graph [17]**). *In the class of Unit Disk Graphs* (UDG) *an edge between two distinct vertices u and v exists iff $d(u, v) \leq 1$*

As an extension of the UDG concept, *quasi units disk graphs* [2, 45] and *directed transmission graphs* are considered. [6, 77, 59] [1]. Quasi unit disk graphs (QUDG(r)) have a maximum communication distance of 1 and a minimum communication distance of $0 < r \leq 1$. Two distinct vertices with a Euclidean distance less or equal r are always connected, i.e. an edge connecting such a vertex pair is required to exist. Two vertices with Euclidean distance greater than 1 are never connected, i.e. it is forbidden that an edge connecting such a vertex pair exists. Any other vertex pairs can be but do not have to be connected. An edge between such a vertex pair, which can exist, but does not have to exist, is an optional edge. So far, quasi unit disk graphs seem to be only studied as undirected graphs. Therefore, when speaking of quasi unit disk graphs, they are assumed to be undirected.

Definition 7 (**Quasi Unit Disk Graph [2, 45]**). *In the class of Quasi Unit Disk Graphs* (QUDG(r)) *an edge between two distinct vertices u and v exists if $d(u, v) \leq r$, an edge does not exist if $d(u, v) > 1$. For $r < d(u, v) \leq 1$ the existence of the edge uv is undetermined.*

In directed transmission graphs DTG(r), every vertex v has an individual maximum communication distance $s(v) \leq r$ for some general $r > 0$.

Definition 8 (**Directed Transmission Graph [55, 6]**). *In the class of Directed Transmission Graphs* (DTG(r)) *a directed edge from u to v between two distinct vertices u and v exists if $d(u, v) \leq s(u)$, where $s(u) \leq r$ is the individual communication radius of the vertex u and r is a global maximal communication radius.*

Directed transmission graphs can be made symmetric by removing all directed edges where the reverse edge is missing, as described in [55, 6] or by adding the missing reverse edges, as it is additionally considered in [6].

[1] (Note, a maximum communication distance of 1 is considered. This is no limitation to other work, where a maximum communication distance $r > 0$ is used instead. Obviously, scaling vertex positions with $1/r$ yields the same model with maximum communication distance 1)

In the following, the defining properties of the discussed graph models are defined in terms of three base graph classes and the two discussed basic graph transformations, i.e., either removing directed edges where the reverse edge is missing or adding the missing reverse edges.

Definition 9 (**The classes** CRG, MinDG, MaxDG)**.** *Let $G = (V, E)$ be a graph and let $d(u, v)$ be any distance function defined on the vertices (e.g. Euclidean distance when vertices are located in the Euclidean space). For an arbitrary, but fixed value of r, where $0 < r \leq 1$, a graph G is in the class of* min disk graphs $\mathsf{MinDG}(r)$, max disk graphs $\mathsf{MaxDG}(r)$, *resp.* connected region graphs CRG *if it satisfies the following corresponding conditions:*

$$\begin{aligned}
\mathsf{MinDG}(r) &\quad \forall u, v \in V : u \neq v \wedge d(u, v) \leq r \Rightarrow E(u, v) \\
\mathsf{MaxDG}(r) &\quad \forall u, v \in V : d(u, v) > r \Rightarrow \neg E(u, v) \\
\mathsf{CRG} &\quad \forall u, v, w \in V : E(u, w) \wedge u \neq v \wedge d(u, v) \leq d(u, w) \Rightarrow E(u, v)
\end{aligned}$$

Definition 10 (**Symmetric super and subgraph**)**.** *Given a graph $G = (V, E)$, the graph transformations $G^+ = (V, E^+)$ and $G^- = (V, E^-)$, denoted as* symmetric supergraph *resp.* symmetric subgraph *are defined by:*

$$\begin{aligned}
E^+ &= \{uv : uv \in E \text{ or } vu \in E\} \\
E^- &= \{uv : uv \in E \text{ and } vu \in E\}
\end{aligned}$$

When a transformation γ is applied on a whole graph class C, i.e γ is applied on each graph contained in C, then the resulting graph class is written as C^γ. The graph class C^γ therefore consists of all graphs G', where a graph $G \in \mathsf{C}$ exists, such that $G' = G^\gamma$

The intersection of two graph classes A and B yields a graph class $\mathsf{C} = \mathsf{A} \cap \mathsf{B}$. For all graphs $C \in \mathsf{C}$ it holds that $C \in \mathsf{A}$ and $C \in \mathsf{B}$. Consequently a graph in class C has to satisfy the logical axioms of both classes. Obviously, the graph class of unit disk graphs can be built from the defined base classes $\mathsf{MinDG}(r)$ and $\mathsf{MaxDG}(r)$, as follows

$$\mathsf{UDG} = \mathsf{MinDG}(1) \cap \mathsf{MaxDG}(1)$$

Together with the two basic graph transformations, the class of quasi unit disk graphs is given by

$$\begin{aligned}
\mathsf{QUDG}(r) &= (\mathsf{MinDG}(r) \cap \mathsf{MaxDG}(1))^- \\
&= (\mathsf{MinDG}(r) \cap \mathsf{MaxDG}(1))^+
\end{aligned}$$

The latter equality holds, since an edge uv is optional iff vu is optional, i.e., $r < d(u, v) = d(v, u) \leq 1$. Obviously, $\mathsf{QUDG}(1)$ is the special case of unit disk graphs UDG.

The class of directed transmission graphs is given by

$$\mathsf{DTG}(r) = \mathsf{MaxDG}(r) \cap \mathsf{CRG}$$

For a given r, this follows immediately by choosing the individual maximum communication distance of a vertex v as $s(v) = \max\{d(v, w) : vw \in E\}$. The symmetric variant considered in [55] has no maximum communication distance and is obtained by removing all directed edges where the reverse edge is missing. It is thus described by CRG^-. The symmetric variants considered in [6] correspond to $\mathsf{DTG}(r)^-$ and $\mathsf{DTG}(r)^+$

2.3.2 Connectivity and Planarity of Graphs

The previously defined graph classes UDG, QUDG and DTG, have in common that the existence of edges is determined by strict bounds on the distances between the corresponding vertices. However, in a realistic WSN it is not very likely that the existence of edges is determined solely by using such strict bounds. Therefore, less stringent graph properties are of interest in order to obtain more realistic graph models. One such property is the condition that edges in a graph are not intersecting. With the following definition, the intersection of edges can be defined in a way, such that also the case of collinear vertices is covered precisely.

Definition 11 (**Intersection of edges).** *Two edges uv and wx are intersecting iff they have a point in common that is not a vertex of both edges and the edges are not equal.*

Based on the definition for intersecting edges a plane graph is defined.

Definition 12 (**Plane graph).** *A graph $G = (V, E)$ is plane iff it does not contain a pair of intersecting edges.*

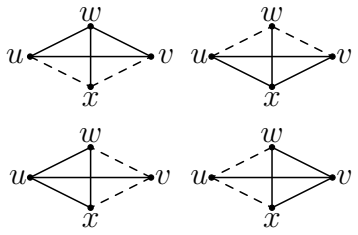
Another important property is the connectivity of graphs. In order to define a connected graph, first a path connecting two vertices u and v is defined.

Definition 13 (**Path).** *A path $p(u, v)$ between two vertices $u, v \in V$ is a sequence of edges $w_iw_{i+1} \in E$ for $0 \leq i \leq k-1$ between pairwise disjoint vertices $w_0, \ldots, w_k \in V$ with $u = w_0$, $v = w_k$. The length of the path corresponds to the number of edges and is therefore k.*

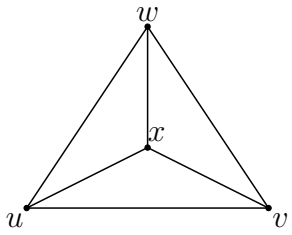
Using the definition of a path, a connected graph is defined.

Definition 14 (**Connected graph).** *A graph $G = (V, E)$ is connected iff for all vertices $u, v \in V$ with $u \neq v$ a path connecting u and v exists.*

If the intersection of two edges was defined without the restriction that the common point of the intersecting edges is not a vertex of both both edges, a connected graph with at least 3 vertices could never be plane. Furthermore, the condition that an edge cannot intersect with itself is required, because otherwise every graph containing an edge would not be plane.



(**a**) Redundancy property.



(**b**) Coexistence Property.

Figure 2: Redundandcy and Coexistence property [11].

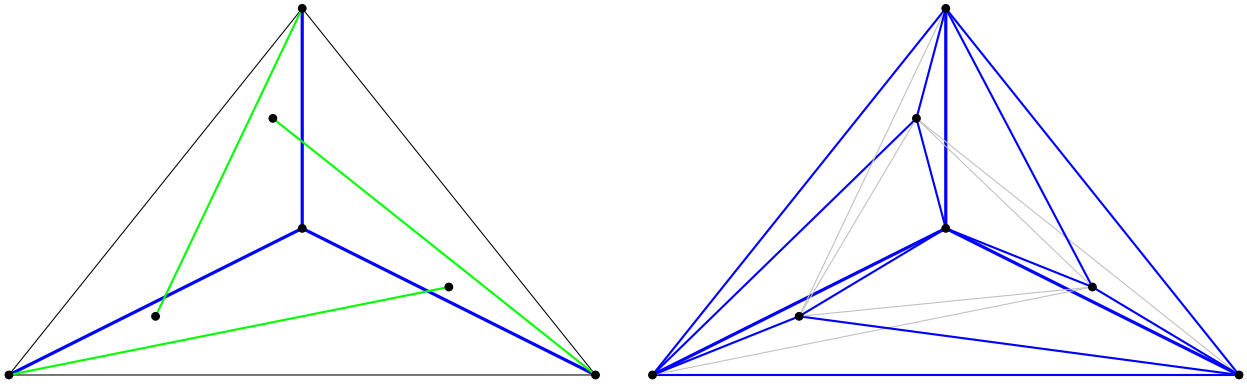
Next, properties related to the intersection of edges and paths connecting vertices are considered. Such properties are for example the redundancy and coexistence property. The redundancy property [32, 62] assures that for at least one edge of a pair of intersecting edges, an alternative path exists that connects the two vertices of this edge.

Definition 15 (**Redundancy Property).** *For a pair of intersecting edges uv and wx the redundancy property is satisfied if among the vertices u, v, w, x at least one vertex is connected by an edge to the other three vertices.*

In order to satisfy the redundancy property for an intersection of the edges uv and wx at least one of the following edge pairs (1) uw and ux, (2) vw and vx, (3) wu and wv or (4) xu and xv (see Fig. 2(**a**)) has to exist. The edges of the edge pairs (1)-(4) are termed as *redundancy edges*. The redundancy edges form an alternative path connecting the vertices u and v resp. w and x. An edge between a vertex pair, which is connected by a path of redundancy edges is termed a *redundant edge*. The redundancy property is also satisfied if three or all four of the edges uw, ux, vw and vx exist. In case that three or all four of the redundancy edges exist, both of the edges uv and wx are redundant edges.
Investigating the dependency between redundancy and redundant edges, it turns out that the redundancy and redundant edges can cyclically depend on each other (see Fig. 3(**a**)). To avoid this cyclic dependency for alternative paths of pairs of intersecting edges, a further property is required (see Fig. 3(**b**)). Such a property can be the coexistence property. The coexistence property [52] assures that vertices inside a triangle are connected with the border vertices of the triangle (see Fig. 2(**b**)).

Definition 16 (**Coexistence Property).** *For a vertex x located inside a triangle formed by the edges uv, vw and wu, also the edges ux, vx and wx exist.*

The definitions for the redundancy and coexistence property are now used to define the class of redundancy and coexistence graphs (RCG).



(**a**) The blue edges are forming a redundancy cycle

(**b**) The coexistence property can be used to avoid a redundancy cycle.

Figure 3: Illustration of a redundancy cycle and a way to avoid a redundancy cycle using the coexistence property

Definition 17 (RCG). *For a given vertex set, a redundancy and coexistence graph (RCG) is a graph G, satisfying the redundancy property for each pair of intersecting edges and the coexistence property for each vertex located inside a triangle. The class* RCG *consists of all possible RCGs for a given vertex set.*

A concept for transforming a connected graph from the class RCG into a connected and plane graph is described in the following subsection.

2.3.3 Graph Transformations and Graph Constructions

In this subsection, some graph classes and graph (class) transformations are defined, which are more general than the graph classes and graph (class) transformation considered so far. The graph classes considered in the following correspond to graph classes which satisfy a certain property, like degree limitation, spanning property or planarity, i.e. the property that edges do not intersect. Furthermore, the graph (class) transformations considered in the following are used to obtain graphs (resp. classes of graphs), which satisfy such properties. Planarity and the construction of plane drawings is well studied, for example in the graph models of Delaunay triangulations [25], Gabriel graphs [30] and relative neighborhood graphs [75]. The corresponding graph classes are considered in the following after defining some basic graph properties.

Definition 18 (Subgraph). *For a given graph $G = (V, E)$ the graph $G' = (V', F)$ is a subgraph of G if $V' \subseteq V$ and $F \subseteq E$*

Definition 19 (Spanning Subgraph). *For a given graph $G = (V, E)$ the graph $G' = (V, F)$ is a spanning subgraph of G if G' contains the same vertex set as G and $F \subseteq E$.*

Definition 20 (**Induced Subgraph).** *For a given graph $G = (V, E)$ and $V' \subseteq V$ the induced subgraph graph $G' = (V', E')$ consists of the vertex set V' and the edge set E' of all edges from E connecting the vertices in V'.*

Definition 21 (**Clique).** *A clique C in a graph $G = (V, E)$ is a subset of vertices $C \subseteq V$, for which the induced subgraph on the vertex set C is a complete graph.*

Remark 1. *A special case of an induced subgraph is a graph G' which is obtained by removing a single vertex v and all corresponding edges incident with v from G. For such a graph the notation $G' = G \setminus v$ is used.*

Remark 2. *A special case of a spanning subgraph is a graph G' which is obtained by removing a single edge from G. For such a graph the notation $G' = G \setminus uv$ is used.*

The construction of (spanning) subgraphs is usually used in the context of this thesis to remove intersections of edges from a given graph.
Another possibility for graph constructions is to add vertices and edges to a given graph. For this task, a terminology for adding edges resp. paths to a given graph is required. Adding an edge uv to a given graph $G = (V, E)$ results in a graph $G' = (V', E')$ with vertex set $V' = V \cup \{u, v\}$ and edge set $E' = E \cup \{uv\}$. For adding an edge the notation $G' = G + uv$ is used. If a path $p(u, v)$ instead of an edge is added, the notation $G' = G + p(u, v)$ is used.
Next, some graph classes are considered, which are used for the construction of a plane and connected graph for a given vertex set. The graph classes are defined using logical notations with predicates for the vertices and edges. In a Gabriel graph, an edge between two vertices u and v exists, iff there exists no other vertex w, which is located inside or on the smallest circle passing through u and v.

Definition 22 (**Gabriel Graph [30]).** *Let $m(u, v)$ be the middle point of the segment uv. In a Gabriel graph (GG) $G = (V, E)$ an edge between u and v exists iff*

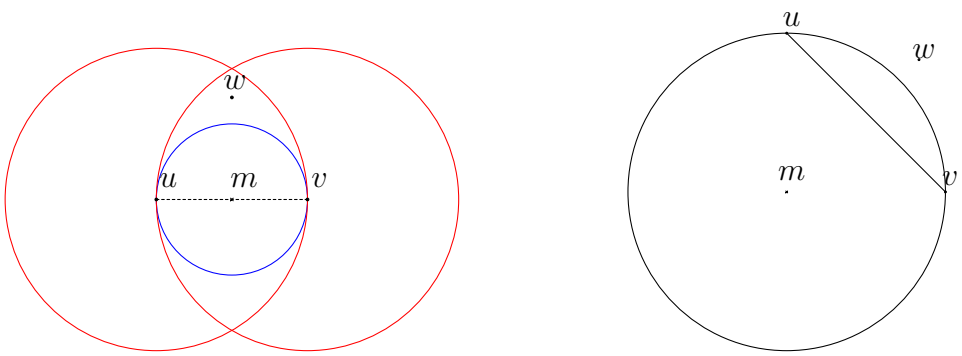
$$\forall w : V(w) \wedge u \neq w \wedge v \neq w \wedge \rightarrow d(u, m(u, v)) < d(w, m(u, v))$$

In a relative neighborhood graph, there exists an edge between the vertices u and v, iff there exists no vertex, w, which is closer to both vertices u and v, than u and v to each other.

Definition 23 (**Relative Neighborhood Graph [75]).** *In a Relative neighborhood graph (RNG) $G = (V, E)$ an edge between u and v exists iff*

$$\forall w : V(w) \rightarrow d(u, v) \leq d(u, w) \vee d(u, v) \leq d(v, w)$$

In a Delaunay triangulation an edge between the vertices u and v exists, if there exists a circle passing through u and v, such that there is no other vertex x located inside that circle.



(**a**) The edge uv exists in the GG, but not in the RNG

(**b**) The edge uv exists in the Delaunay triangulation

Figure 4: Illustration of the conditions for the existence of edges in a Gabriel graph, relative neighboorhood graph and a Delaunay triangulation

Definition 24 (Delaunay Triangulation [25]). *In a Delaunay Triangulation (DT) an edge between u and v exists if*

$$\exists m : d(u,m) = d(v,m) \wedge \forall x : \Big(\big(V(x) \wedge u \neq x \wedge v \neq x \big) \rightarrow d(m,x) > d(u,x) \Big)$$

The conditions for the existence of an edge uv in a Gabriel graph, relative neighborhood graph resp. a Delaunay triangulation are illustrated in Fig.4.
In contrast to the Gabriel graph and the relative neighborhood graph the condition for the existence of edges in the Delaunay triangulation cannot be defined by a "if and only if" condition in case there exist cocircular vertices, since in case there exist cocircular vertices, the Delaunay triangulation is not clearly defined. This means, that only in case that there exists no cocircular vertices, the above condition can be used as an "if and only if" condition.

So far it was assumed, that for all vertices all information about the other vertices and edges are available. In the next subsection it is considered, how graphs can be constructed resp. transformed into other graphs, if only information about vertices and edges that are "close" to a vertex are available

2.3.4 Locality in Graphs

In order to make computations more efficient, an important concept for the transformation of a source graph into an image graph with certain properties, is locality. A property $P(v_1, \ldots, v_n)$ of vertices is local if it only depends on the vertices and edges in a neighborhood of the vertices $v_1, \ldots, v_n$. Locality can be studied in different variants, depending on the definition of the neighborhood of a vertex. An overview about the concept of locality and the local execution of algorithms can be found in [58].

The most commonly used concept of locality is considered in [49]. In this concept the neighborhood $N_1(v)$ of a vertex v consists of v and its direct neighbors, while the k-hop neighborhood $N_k(v)$ of v consists of all vertices connected to v in at most k steps. This approach of locality is used for example, when properties of a graph in the class of RCG are considered.
An alternative approach is to consider vertices in a geometric neighborhood of v. This approach is used in [16], where a property is defined to be local if it is expressed as a function of the vertices geometrically close to the studied ones. This approach of locality based on a geometric neighborhood can be used for studying graph class like UDG, QUDG, DTG and related graph classes, as well as for studying graph classes like GG, RNG and DT.

In a local algorithm only the information about vertices and edges in a certain neighborhood is available to take a decision for a certain vertex. The advantage of a local algorithm is that only the relevant subset of information of the graph is considered, which reduces the running time. A further advantage is that local algorithms can be implemented in a distributed version, where many vertices take their decision in parallel. However, a common challenge for distributed algorithms is, how to deal with conflicting decisions of different vertices at the same time [35]. Typically a global ordering is applied in this case to break the ties.

2.4 Provers

In this section the tools for performing the proof tasks are briefly described.

For checking the satisfiability of formulae over one or more theories, the theorem prover **Z3** [23], devolved by Microsoft Research, is used. Beside checking the satisfiability of a formula, Z3 can also be used to compute a model in the case that a formula is satisfiable. Furthermore, in the case that Z3 detects the unsasifiablity of a formula, it can be used to compute an unsat core.

For hierarchical proving in local theory extensions the program **H-PILoT** [39], developed in the research group of Prof. Dr Sofronie-Stokkermans, is used. H-PILoT takes as input a problem in a theory extensions or a chain of theory extensions, and reduces it to a problem in a base theory. The satisfiability of the reduced problem in the base theory is afterwards tested with a SMT solver, like Z3. An alternative SMT solver that can be used is **Redlog**, an extension of the computer algebra system reduce to a logical system allowing quantifier elimination.

A further tool used for handling proof tasks is **Mathematica**, a software system by Wolfram with built-in libraries used for symbolic computation, visualization and numerical evaluations and many more application domains.

3 Graph Class Containedness Relations Between Graph Classes Related to Planarity Conditions

The concepts Gabriel graph, relative neighborhood graph and Delaunay triangulation are used to construct connected and plane graphs for a fixed vertex set in the Euclidean plane. A Delaunay triangulation is a maximal plane spanning graph, i.e. no edge can be added without loosing the planarity of the graph. On the other hand every subgraph of a Delaunay triangulation is still a plane graph. Since a Gabriel graph contains a subset of the edges of a Delaunay triangulation and a relative neighborhood graph contains a subset of the edges of a Gabriel graph, the graphs constructed by both concepts are also plane drawings. However, these relations are not guaranteed to hold anymore, if another distance function than the Euclidean distance is used. It is therefore investigated for which distance functions the condition that the edges of one graph construct are a subset of edges of another graph construct still holds. Furthermore, it is investigated which of the graph concepts are still guaranteed to be plane graphs. Since the Delaunay triangulation is more complicated to model than the Gabriel graph and the relative neighborhood graph, the analysis is restricted to the two later graph concepts.
Parts of the results presented in this section were already published in [13]

3.1 Graph Classes Related to Planarity Conditions

The goal in this section is to express the conditions of geometric graph classes by axioms. These axioms correspond to conditions, which describe when certain edges exist and when the edges do not exist. After describing the graph classes by such axioms it is investigated, if a graph class C_1 is contained in a graph class C_2. If C_1 is contained in C_2, then every graph satisfying the axioms of C_1, also satisfies the axioms of C_2. On the other hand if C_1 is not contained C_2, then there exists a graph satisfying the axioms of the graph class C_1, but not the axioms of the graph class C_2.

3.1.1 Axiomatic Description of the Graph Classes

In the following, undirected graphs without self-loops are considered. Such properties can be described by the axioms $\mathcal{K}_0$:

$\mathcal{K}_0 = \{\forall x(\neg E(x,x)), \quad \forall x,y\, E(x,y) \to E(y,x), \quad \forall x,y\, E(x,y) \to V(x) \wedge V(y)\}$.

The investigated graph classes are the classes of Gabriel graphs (GG), relative neighborhood graphs (RNG) and plane drawings (P).

The class GG of Gabriel graphs [15] is axiomatized by $\mathcal{K}_0$ together with:

(GG) $\quad \forall u,v\ E(u,v) \leftrightarrow \pi_{\mathsf{GG}}(u,v)$

where $\pi_{\mathsf{GG}}(u,v)$ expresses the condition that u and v are vertices and every vertex other than u and v lies outside of the minimal circle passing through u and v. Let $m(u,v)$ be the middle point of the segment uv, i.e $d(u,v) = 2 \cdot d(m(u,v),u) = 2 \cdot d(m(u,v),v)$.

Then the predicate $\pi_{\mathsf{GG}}(u,v)$ is defined by:

$$\begin{aligned}\pi_{\mathsf{GG}}(u,v) := \;& V(u) \wedge V(v) \wedge u \neq v \wedge \\ & \forall w \big(w \neq u \wedge w \neq v \wedge V(w) \rightarrow d(m(u,v),w) > d(m(u,v),u)\big).\end{aligned}$$

The superclasses $\mathsf{GG}^{\rightarrow}$ and $\mathsf{GG}^{\leftarrow}$ of GG are defined by only keeping one of the implications in (GG). The superclass $\mathsf{GG}^{\rightarrow}$ is defined by only keeping the "$\rightarrow$" implication of (GG), while $\mathsf{GG}^{\leftarrow}$ is defined by only keeping the "$\leftarrow$" implication. Therefore, $\mathsf{GG}^{\rightarrow}$ contains a subset of the edges of GG, while $\mathsf{GG}^{\leftarrow}$ contains a superset of the edges of GG.

The class RNG of relative neighborhood graphs [42] is axiomatized by $\mathcal{K}_0$ together with

(RNG) $\qquad \forall u,v \; E(u,v) \leftrightarrow \pi_{\mathsf{RNG}}(u,v)$

where $\pi_{\mathsf{RNG}}(u,v)$ expresses the condition that there exists no vertex, which is closer to both vertices u and v, than u and v to each other.

The predicate $\pi_{\mathsf{RNG}}(u,v)$ is defined by:

$$\pi_{\mathsf{RNG}}(u,v) := V(u) \wedge V(v) \wedge u \neq v \wedge \forall w \, (V(w) \rightarrow d(u,w) \geq d(u,v) \vee d(w,v) \geq d(u,v)).$$

Also for the class of RNG the superclasses $\mathsf{RNG}^{\rightarrow}$ and $\mathsf{RNG}^{\leftarrow}$ are defined. The superclass $\mathsf{RNG}^{\rightarrow}$ is defined by only keeping the "$\rightarrow$" implication in (RNG), while $\mathsf{RNG}^{\leftarrow}$ is defined by only keeping the "$\leftarrow$" implication.

The class P of plane drawings can be described using the axiom:

(P) $\qquad \forall u,v,w,x : E(w,x) \wedge \pi_P(u,v,w,x) \rightarrow \neg E(u,v)$

where $\pi_P(u,v,w,x)$ expresses a necessary condition that the segments uv and wx [2] intersect. This can either be expressed by using analytic geometry or (if d is a arbitrary distance function) by the following formula:

$$\begin{aligned}\pi_{\mathsf{P}}(u,v,w,x) \;:=\; & V(u) \wedge V(v) \wedge V(x) \wedge V(w) \wedge u \neq v \wedge w \neq x \wedge \\ & \exists q (d(u,q) + d(q,v) = d(u,v) \wedge d(w,q) + d(q,x) = d(w,x))\end{aligned}$$

For some distance functions, like the Euclidean distance, the predicate $\pi_P(u,v,w,x)$ expresses also a sufficient condition for the intersection of segments. However, for other distance functions $\pi_P(u,v,w,x)$ can be true, but the according segments uv and wx do not intersect. Therefore, models which disprove the planarity of a graph have to be investigated more closely and if nescassery additional constraints have to be used.

The graph classes are modeled using a logical language and consider extensions of the theory of real closed fields and of theories modeling the points with additional functions symbols (V, E, distances, etc.). In [61] it was proven that the properties of theory extensions occurring in this context can be used to define sound, complete and terminating proof procedures.

[2] For the existence of these segments it is necessary that $u \neq v$ and $w \neq x$. If, for instance, the vertex w is located on the segment uv, but not equal to u or v, then the segments uv and wx intersect.

3.1.2 Conditions for Testing Class Containedness Relations

In [13] second-order quantifier elimination was used to determine constraints on abstract predicates used in the description of the classes under which containdness relations hold. With these constraints, testing class containdeness relations for geometric graph classes is possible.

Theorem 2 ([13]). *Given the graph classes C_i, $C_i^{\rightarrow}$ and P defined by:*

$$\begin{array}{lllll} \mathsf{Ax}_{\mathsf{C_i}} & : & \forall u, v & E(u,v) \leftrightarrow \pi_{\mathsf{C_i}}(u,v) & i = 1,2 \\ \mathsf{Ax}_{\mathsf{C_i}^{\rightarrow}} & : & \forall u, v & E(u,v) \rightarrow \pi_{\mathsf{C_i}}(u,v) & i = 1,2 \\ \mathsf{Ax}_{\mathsf{P}} & : & \forall u, v, w, x & E(w,x) \wedge \pi_{\mathsf{P}}(u,v,w,x) \rightarrow \neg E(u,v). & \end{array}$$

For class containedness relations between the classes $\mathsf{C_i}$, $\mathsf{C_i}^{\rightarrow}$ and P it holds that

$$\mathsf{C}_1 \subseteq \mathsf{C}_2 \quad \textit{iff} \quad (\pi_{\mathsf{C}_1}(a,b) \wedge \neg\pi_{\mathsf{C}_2}(a,b)) \vee (\neg\pi_{\mathsf{C}_1}(a,b) \wedge \pi_{\mathsf{C}_2}(a,b)) \textit{ is unsatisfiable} \tag{1}$$

$$\mathsf{C}_1^{\rightarrow} \subseteq \mathsf{C}_2^{\rightarrow} \quad \textit{iff} \quad \pi_{\mathsf{C}_1}(a,b) \wedge \neg\pi_{\mathsf{C}_2}(a,b) \textit{ is unsatisfiable} \tag{2}$$

$$\mathsf{C}_1 \subseteq \mathsf{P} \quad \textit{iff} \quad \pi_{\mathsf{P}}(a,b,c,d) \wedge \pi_{\mathsf{C}_1}(c,d) \wedge \pi_{\mathsf{C}_1}(a,b) \textit{ is unsatisfiable} \tag{3}$$

$$\mathsf{C}_1^{\rightarrow} \subseteq \mathsf{P} \quad \textit{iff} \quad \mathsf{C}_1 \subseteq \mathsf{P} \tag{4}$$

A detailed manual proof of Theorem 2 can be found in [60].

3.2 Locality of Theory Extensions Involving Distances

In order to test the class containedness relations, the theories used for formalizing points, distance and vertices have to be investigated. The concrete form of such a theory $\mathcal{T}$ depends on the model for the considered set of points and distances. In order to define such a theory, combinations and extensions of theories are used. A combination of theories is a new theory formed by the union of the axioms and the signature of previously defined theories. An extension of a theory is a new theory obtained by adding function symbols, predicate symbols and axioms, which formalize properties of the new symbols, to a previously defined theory. For modeling the Euclidean plane with the Euclidean distance, $\mathcal{T}$ can be defined starting from the combination of a theory of points P over $\mathbb{R} \times \mathbb{R}$ and the theory $\mathbb{R}$ of real closed fields, using a chain of theory extensions:

$$(E) \qquad P \cup \mathbb{R} \;\subseteq\; P \cup \mathbb{R} \cup \mathsf{Free}_{x,y} \;\subseteq\; \mathcal{T} = P \cup \mathbb{R} \cup \mathsf{Free}_{x,y} \cup \mathcal{T}_d^e$$

The theory $\mathsf{Free}_{x,y}$ is the free theory, where x and y are unary functions, which do not have any additional properties. The functions x and y are used as the coordinate functions. $\mathcal{T}_d^e$ is the axiom stating that the Euclidean distance between two points u and v is $d(u,v) = \sqrt{(x(u)-x(v))^2+(y(u)-y(v))^2}$. However, the square root is not part of the signature, since only the squared distance is $d(u,v)^2$ is considered in the according tests.

In a more general context, $\mathcal{T}_d^n$ is the axiom stating that the distance function between two points u and v is a distance function induced by the p-(quasi-)norm $\|x\|^p$, which is defined as $d(u,v) = (|x(u)-x(v)|^p+|y(u)-y(v)|^p)^{1/p}$ for $p > 0$.

For $p \geq 1$, $\|x\|^p$ is a norm and the induced distance function is a metric, where $p = 2$ corresponds to the Euclidean distance. The metric for $p = 1$ is called the Manhattan metric and the metric for $p = \infty$ is called the maximum metric. The distance function for the maximum metric is defined as $d(u,v) = max(|x(u) - x(v)|, |y(u) - y(v)|)$.

For $1 < p < \infty$ the condition $d(u,v) = d(u,w) + d(w,v)$ expresses the colinearity condition, i.e. u, v and w are located on a straight-line segment. However, for $p = 1$ and $p = \infty$ there exist also points which are not located on a straight-line segment, satisfying $d(u,v) = d(u,w) + d(w,v)$. Therefore models for potential intersections for $p = 1$ and $p = \infty$ must be carefully analyzed to verify whether the corresponding segments actually intersect.

For $0 < p < 1$ $\|x\|^p$ is not a norm, but only a quasinorm, since the triangle inequality is not satisfied anymore. Therefore, the induced distance function is not a metric anymore in this case.

In order to model the two dimensional plane with a distance function induced by the p-(quasi-)norm, $\mathcal{T}$ can be defined in a similar way as the theory considered for the Euclidean plane with the Euclidean distance, using a chain of theory extensions:

$$(N) \qquad P \cup \mathbb{R} \subseteq P \cup \mathbb{R} \cup \mathsf{Free}_{x,y} \subseteq \mathcal{T} = P \cup \mathbb{R} \cup \mathsf{Free}_{x,y} \cup \mathcal{T}_d^n$$

Another possibility is to use distance functions, which are not using coordinates. For modeling arbitrary metric spaces (X,d), the following axioms have to be satisfied by the distance function d.

$$\begin{array}{lll}
(d_1) & \forall x,y & d(x,y) \geq 0 \\
(d_2) & \forall x,y,z & d(x,y) \leq d(x,z) + d(z,y) \\
(d_3) & \forall x,y & d(x,y) = d(y,x) \\
(d_4) & \forall x,y & x = y \rightarrow d(x,y) = 0 \\
(d_5) & \forall x,y & d(x,y) = 0 \rightarrow x = y
\end{array}$$

In this case $\mathcal{T}$ can be defined, using a chain of theory extensions

$$(M) \qquad P \cup \mathbb{R} \subseteq \mathcal{T} = P \cup \mathbb{R} \cup \mathcal{T}_d^m$$

where P is the theory of pure equality, $\mathbb{R}$ the theory of real closed fields and $\mathcal{T}_d^m$ are the axioms $d_1 - d_5$ of a metric. The theory of pure equality has no function symbols, equality as the only predicate symbol and contains no non-logical axioms.

For quasi-metric spaces (X,d), the triangle inequality d_2 is replaced by a relaxed version

$$(d_2^K) \quad \forall x,y,z \quad d(x,y) \leq K \cdot (d(x,z) + d(z,y)) \quad \text{for} \quad K \geq 1$$

In this case $\mathcal{T}$ is defined in a similar way as for a metric space, by using a chain of theory extensions:

(Q) $\quad P \cup \mathbb{R} \subseteq \mathcal{T} = P \cup \mathbb{R} \cup \mathcal{T}_d^q$

where $\mathcal{T}_d^q$ are the axioms d_1 and d_3-d_5 of a metric and d_2^K.

Most of these extensions are known to be local. In (E) and (N) there exist extensions with free function symbols and definitional extensions [69]. In [61] it was proven that the axioms of a metric define a Ψ-local extension of $P \cup \mathbb{R}$ with a closure operator Ψ. Since d_2 is a special case of d_2^K, it is likely, that the axioms d_1 and $d_3 - d_5$ of a metric also define Ψ-local extension of $P \cup \mathbb{R}$ with a closure operator Ψ. A proof is planed for future work. HPiLoT is used for proof tasks and for counterexamples. Since, HPiLoT is guaranteed to always find correct counterexamples only for local theory extensions, the obtained models were carefully analyzed. It turned out, that the counterexamples are correct.

3.3 Class Containedness Relations for Different Distance Functions

The class containedness relations for the classes GG, RNG and P are now investigated for different distance functions. Theorem 2 states that it is enough to consider a graph class $\mathsf{C}^{\rightarrow}$ instead of C to determine whether C is contained in the class of plane drawings P. Therefore, the graph classes $\mathsf{RNG}^{\rightarrow}$ and $\mathsf{GG}^{\rightarrow}$ are regarded instead of RNG and GG.
The class containedness relations that are considered are therefore:

(a) $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$

(b) $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{RNG}^{\rightarrow}$

(c) $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$

(d) $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{P}$

Since the graph class $\mathsf{C}^{\leftarrow}$ contains all graphs with a superset of edges of C, the class $\mathsf{C}^{\leftarrow}$ cannot be used for testing if a graph class C is contained in the class of plane drawings P.

In [13] the class containedness relations (a)-(d) were considered for d as the Euclidean distance and for an arbitrary metric. The results are presented in Section 3.3.1.

In [68] the class containedness relations (a) and (b) were studied for the Manhattan and maximum metric. Those results are presented in Section 3.3.2. Afterwards the remaining class containedness relations are considered for the Manhattan and maximum metric.

In Section 3.3.3 the class containedness relations (a)-(d) are considered for a metric induced by a p-norm with $1 < p < \infty$. H-PILoT was used to test some cases, where p is a rational number. For this kind of tests H-PILoT did not terminate, when Z3 was used as an end prover. When redlog was used as an end prover in some cases a result was obtained after a long running time (often more than one hour). However, it was not possible to obtain models, when redlog was used as prover. Since p can be an arbitrary positive real number, a general proof was therefore not possible with H-PILoT and is instead given manually.

Finally in Section 3.3.4 the class containedness relations (a)-(d) are considered for distances functions, which are not satisfying the triangle inequality. Such distance functions are not a metric anymore and therefore results for class containedness relations obtained for a general metric cannot be used anymore. One considered distance function is the distance function induced by the p-quasinorm for $0 < p < 1$. Another distance function that is considered, is a distance function where the axioms for a metric are considered, but with a relaxed version of the triangle inequality. For the second kind of distance function H-PILoT could be used to test the class containedness relations, while for the distance functions induced by a p-quasinorm a general proof using H-PILoT was not possible.

3.3.1 Euclidean Distance and Arbitrary Metric

The results from [13] for testing the containedness relations (a)-(d) are presented in the following, first for the Euclidean distance and afterwards for an arbitrary metric. If necessary, in all following cases the coordinates of the middle point m of u and v are computed as

$x(m) = (x(u) + x(v))/2$
$y(m) = (y(u) + y(v))/2.$

Furthermore, in order to guarantee that all considered points are different, the following conditions are added in all following cases: $x(r) \neq x(s) \vee y(r) \neq y(s)$ for $r, s \in \{u, v, w, m\}$.

Case 1: d is the Euclidean distance.

(a) To prove that $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ holds, it is sufficient to show that $\pi_{\mathsf{RNG}}(u, v) \wedge \neg\pi_{\mathsf{GG}}(u, v)$ is unsatisfiable. The formula obtained from $\neg\pi_{\mathsf{GG}}(u, v)$ after skolemization and simplification is a ground formula.

H-PILoT derives unsatisfiability, i.e. proves that $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$.

(b) For proving that $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$ it is shown that $\pi_{\mathsf{GG}}(u, v) \wedge \neg\pi_{\mathsf{RNG}}(u, v)$ is satisfiable. H-PILoT gives the answer satisfiable and produces the following model:

$u = (1, 0), \quad v = (7, -8), \quad w = (-1, -5), \quad m = (4, -4).$

(c) and **(d)** Since $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ holds for the Euclidean distance, proving $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ also implies $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{P}$. The containedness relation $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ holds iff $\pi_{\mathsf{P}}(u, v, w, x) \wedge \pi_{\mathsf{GG}}(w, x) \wedge \pi_{\mathsf{GG}}(u, v)$ is unsatisfiable. With the encoding for the Euclidean distance H-PILoT did not terminate (after more than one hour).

Case 2: d is an arbitrary metric.

(a) For testing the satisfiability of $\pi_{\mathsf{RNG}}(u, v) \wedge \neg\pi_{\mathsf{GG}}(u, v)$ H-PILoT answers satisfiable and returns the following model (impossible in the Euclidean space):

$d(u, v) = 6, \ d(u, m) = 3, \ d(v, m) = 3, \ d(u, w) = 6, \ d(v, w) = 5, \ d(w, m) = 3.$

Thus, the containedness relation $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ is true for the Euclidean distance, but not for an arbitrary metric.

(b) Since $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$ for the Euclidean distance, this also holds for an arbitrary metric.

(c) For testing the satisfyiability of $\pi_{\mathsf{P}}(u,v,w,x) \wedge \pi_{\mathsf{GG}}(w,x) \wedge \pi_{\mathsf{GG}}(u,v)$, H-PILoT returns "unsatisfiable". This proves that $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ is true.

(d) For an arbitray metric, it holds that $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{P}$. This is proven by deriving graphs in $\mathsf{RNG}^{\rightarrow}$, which are not in P for the Manhattan and maximum metric in Section 3.3.2.

Containedness relations derived by combing the results of Case1 and Case2
For the Euclidean distance it holds that $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$. Furthermore, for an arbitrary metric it holds that $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$. In total, the following containedness relations hold for the Euclidean distance: $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$. Hence, condition (4) form Theorem 2 yields $\mathsf{GG} \subseteq \mathsf{P}$ and $\mathsf{RNG} \subseteq \mathsf{P}$ for the Euclidean distance.

3.3.2 Manhattan and Maximum Metric

For the Manhattan and the maximum metric the containdness relations (a) and (b) were tested in [68]. For both metrics counterexamples were constructed, which prove that $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{GG}^{\rightarrow}$ and $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$. The containedness relations (c) does not have to be investigated separately, since $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ for an arbitrary metric. However, the containedness relation (d) has to be investigated separately, since $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{GG}^{\rightarrow}$ for the Manhattan or the maximum metric. The details for testing the containedness relations (a)-(d) are presented in the following, first for the Manhattan metric, afterwards for the Maximum metric.

Case 3: d is the Manhattan metric.
(a) ([68]) For proving that $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{GG}^{\rightarrow}$ it is shown that $\pi_{\mathsf{RNG}}(u,v) \wedge \neg\pi_{\mathsf{GG}}(u,v)$ is satisfiable. H-PILoT gives the answer satisfiable and produces the following model:

$$u = (-14,-2), \quad v = (0,0), \quad w = (-11,-5), \quad m = (-7,-1).$$

The reason for $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{GG}^{\rightarrow}$ is that a unit disk for the Manhattan metric is a rectangle and therefore for vertices located on the border of such a unit disk the condition for including the edge in $\mathsf{RNG}^{\rightarrow}$ is satisfied, while the condition for including the edge in $\mathsf{GG}^{\rightarrow}$ is not satisfied.

(b) ([68]) For proving that $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$ it is shown that $\pi_{\mathsf{GG}}(u,v) \wedge \neg\pi_{\mathsf{RNG}}(u,v)$ is satisfiable. H-PILoT gives the answer satisfiable and produces the following model:

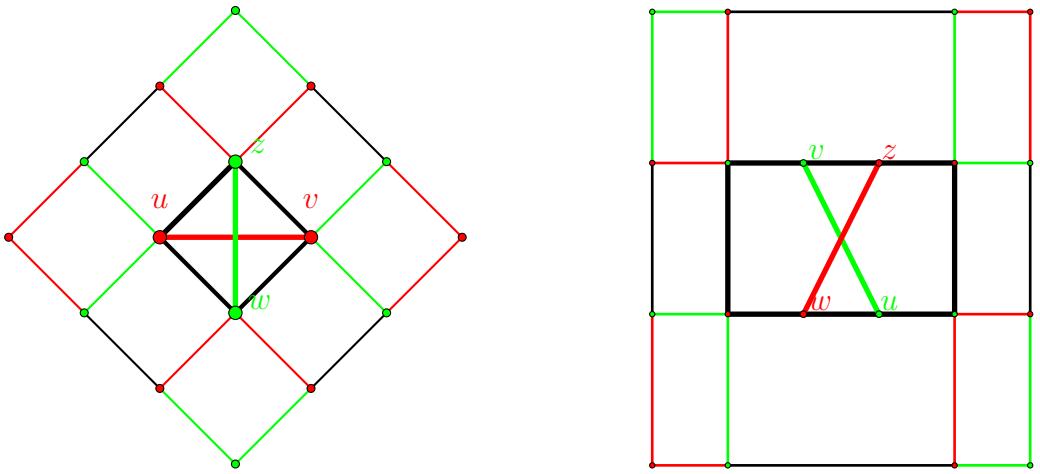
$$u = (12,-4), \quad v = (0,4), \quad w = (-11,0), \quad m = (6,0).$$

(c) $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ holds for an arbitrary metric and therefore also for the Manhattan metric.

(d) For proving that $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{P}$ it is shown that $\pi_{\mathsf{P}}(u,v,w,x) \wedge \pi_{\mathsf{RNG}}(w,x) \wedge \pi_{\mathsf{RNG}}(u,v)$ is satisfiable. H-PILoT gives the answer satisfiable and produces the following model (Fig. 5(**a**)):

$u = (0,1)$, $v = (2,1)$, $w = (1,0)$ and $z = (1,2)$.

The distance between any pair of two different vertices is 2. Therefore, the edges uv and wz exist in the relative neighborhood graph. Moreover, the segments uv and wz intersect in the point $q = (1,1)$.



(**a**) For the Manhattan metric $\mathsf{RNG} \not\subseteq \mathsf{P}$ (**b**) For the maximum metric $\mathsf{RNG} \not\subseteq \mathsf{P}$

Figure 5: Counterexamples proving $\mathsf{RNG} \not\subseteq \mathsf{P}$ for the Manhattan and the maximum metric

Case 4: d is the maximum metric.

(a) ([68]) For proving that $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{GG}^{\rightarrow}$ it is shown that $\pi_{\mathsf{RNG}}(u,v) \wedge \neg\pi_{\mathsf{GG}}(u,v)$ is satisfiable. H-PILoT gives the answer satisfiable and produces the following model:

$$u = (0,-2), \quad v = (-4,0), \quad w = (0,0), \quad m = (-2,-1).$$

The reason for $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{GG}^{\rightarrow}$ is that a unit disk for the maximum metric is a rectangle and therefore for vertices located on the border of such a unit disk the condition for including the edge in $\mathsf{RNG}^{\rightarrow}$ is satisfied, while the condition for including the edge in $\mathsf{GG}^{\rightarrow}$ is not satisfied.

(b) ([68]) For proving that $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$ it is shown that $\pi_{\mathsf{GG}}(u,v) \wedge \neg\pi_{\mathsf{RNG}}(u,v)$ is satisfiable. H-PILoT gives the answer satisfiable and produces the following model:

$$u = (12,20), \quad v = (0,0), \quad w = (-5,7), \quad m = (6,10).$$

(c) $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ holds for an arbitrary metric and therefore also for the maximum metric.

(d) For proving that $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{P}$ it is shown that $\pi_{\mathsf{P}}(u,v,w,x) \wedge \pi_{\mathsf{GG}}(w,x) \wedge \pi_{\mathsf{GG}}(u,v)$ is satisfiable. H-PILoT gives the answer satisfiable and produces the following model (Fig. 5(**b**)):

$$u = (-6,-8),\ v = (-8,-4),\ w = (-8,-8) \text{ and } z = (-6,-4).$$

The distances are $d(u,v) = d(u,z) = d(w,v) = d(w,z) = 4$ and $d(u,w) = d(v,z) = 2$. Therefore, the edges uv and wz exist in the relative neighborhood graph. Moreover, the segments uv and wz intersect in the point $q = (-7,-6)$.

3.3.3 Distances Induced by Arbitrary p-norms

Distance functions induced by a p-norm were already investigated for the values $p = 1$ (Case3), $p = 2$ (Case1) and $p = \infty$ (Case4) in the previous subsections. In this subsection distance

functions induced by an arbitrary p-norm are considered. The possibilities of using H-PILoT for such a general distance function are quite limited for larger values of p, since the running time becomes too large. The reason for the long running time is the complexity of decision procedures for the theory of real closed fields, especially when large degrees are considered. Therefore, manual proofs for testing the containedness relations are required.

For testing the containedness relations (a) and (b) it turns out that $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ holds for any distance function induced by a p-norm for $1 < p < \infty$, while $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$ for any distance function induced by a p-norm. The containedness relation (c) does not have to be considered separately, since $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ for an arbitrary metric. Also the containedness relation (d) does not have to be considered separately, since $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ and $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ implies $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{P}$. The details for testing the containedness relations (a)-(d) are presented in the following

Case 5: d is an arbitrary distance function induced by a p-norm for $1 < p < \infty$.

(a) To prove that $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ holds, it is shown that $\pi_{\mathsf{RNG}}(u, v) \wedge \neg\pi_{\mathsf{GG}}(u, v)$ is unsatisfiable.

Theorem 3. *The containedness relation $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ holds for any distance function induced by a p-norm with $1 < p < \infty$.*

Proof. The relation $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ holds if for all $u, v, w \in \mathbb{R}^2$ the following condition is satisfied

$$d(u, w) \geq d(u, v) \rightarrow d(m(u, v), w) > d(m(u, v), v) \vee w = v \tag{5}$$

In (5) $m(u, v)$ is the middle point of u and v and $d(u, v)$ the distance function induced by the p-norm. This distance function is defined as $d(u, v) = \left((|x(u) - x(v)|)^p + (|y(u) - y(v)|)^p\right)^{(1/p)}$.

Since shifting all points by a constant value in the same direction does not change the distances, $x(m(u, v))$ and $y(m(u, v))$ can be set to 0 without loss of generality. Furthermore, the p-norm is homogeneous and invariant with respect to reflection on the x-axis and y-axis. Therefore, $d(u, m(u, v))$ can be set to 1, where $x(m(u, v)) - x(u) \geq 0$ and $y(m(u, v)) - y(u) \geq 0$ without loss of generality. This means that $x(u) \leq 0$ and $y(u) \leq 0$ and consequently $x(v) = -x(u) \geq 0$ and $y(v) = -y(u) \geq 0$.
By contraposition, the equation (5) holds if for all w, except $w = v$

$$d(m(u, v), w) \leq d(m(u, v), v) = 1 \rightarrow d(u, w) < d(u, v) = 2 \tag{6}$$

Therefore, an arbitrary, but fixed w can be chosen, such that $d(m(u, v), w) \leq 1$. For simplification in the following formulae $i := x(v)$, $j := y(v)$, $k := x(w)$ and $l := y(w)$ are used

from now on. Since $i \geq 0$, $j \geq 0$ and $d(m(u,v),v) = 1$ it follows that

$$\begin{aligned} & ((|i-0|^p) + |j-0|^p)^{(1/p)} = 1 \\ \Leftrightarrow \quad & i^p + j^p = 1 \\ \Leftrightarrow \quad & j = (1-i^p)^{(1/p)} \end{aligned} \tag{7}$$

In order to prove that (6) holds for an arbitrary, but fixed $w = (k,l)$ with $d(m(u,v),w) \leq 1$, the points $w_1 = (k,l_1)$ and $w_2 = (k,l_2)$ with $d(m(u,v),w_1) = 1$ and $d(u,w_2) = 2$ are considered. Replacing $d(m(u,v),v)$ by $d(m(u,v),w_1)$ and $d(u,v)$ by $d(m(u,v),w_2)$ in (6) yields

$$d(m(u,v),w) \leq d(m(u,v),w_1) = 1 \rightarrow d(u,w) < d(u,w_2) = 2 \tag{8}$$

For proving (8) the (non-negative) values of l_1 and l_2 are computed in the following, depending on the value of k. Since w is chosen such that $d(m(u,v),w) \leq 1$, it holds that $|k| \leq 1$ and $|l| \leq |l_1| \leq 1$. If $k < 0$ or $l < 0$, then $w' = (|k|,|l|)$ can be considered instead of w, since $d(m(u,v),w) = d(m(u,v),w')$ and $d(u,w) \leq d(u,w')$. Therefore, proving (8) for w' is enough to prove it for an arbitrary value of w. So it can be assumed from now on that $k \geq 0$ and $l \geq 0$.

For $k = i$ it holds that $l = l_1(k) = l_2(k) = j$, since $v = w = w_1 = w_2$. For $k \neq i$ it is now proven that $l_2(k) > l_1(k)$ for all $0 \leq k \leq 1$ with $0 \leq l_1(k) \leq 1$. In the same way as j is computed in (7) the value of $l_1(k)$ can be derived as:

$$l_1(k) = (1-k^p)^{(1/p)} \tag{9}$$

From the expression for $d(u,w_2) = 2$ the (non-negative) value of $l_2(k)$ is computed:

$$\begin{aligned} & (|i+k|^p + |l_2(k)+j|^p)^{(1/p)} = 2 \\ \Leftrightarrow \quad & ((i+k)^p + (l_2(k)+j)^p)^{(1/p)} = 2 \\ \Leftrightarrow \quad & (i+k)^p + (l_2(k)+j)^p = 2^p \\ \Leftrightarrow \quad & (l_2(k)+j)^p = 2^p - (i+k)^p \\ \Leftrightarrow \quad & (l_2(k)+j) = (2^p - (i+k)^p)^{(1/p)} \\ \Leftrightarrow \quad & l_2(k) = (2^p - (i+k)^p)^{(1/p)} - j \\ \Leftrightarrow \quad & l_2(k) = (2^p - (i+k)^p)^{(1/p)} - (1-i^p)^{(1/p)} \end{aligned} \tag{10}$$

Let $t := i - k$, i.e. $k = i - t$, where $t \neq 0$ for $i \neq k$. Then $l_2(k) - l_1(k)$ is given by:

$$
\begin{aligned}
& (2^p - (i + (i-t))^p)^{(1/p)} - (1 - i^p)^{(1/p)} - (1 - (i-t)^p)^{(1/p)} \\
= \; & (2^p - (2i - t)^p)^{(1/p)} - (1 - i^p)^{(1/p)} - (1 - (i-t)^p)^{(1/p)} \\
= \; & (2^p - (2 \cdot (i - t/2))^p)^{(1/p)} - (1 - i^p)^{(1/p)} - (1 - (i-t)^p)^{(1/p)} \\
= \; & (2^p - 2^p \cdot (i - t/2)^p)^{(1/p)} - (1 - i^p)^{(1/p)} - (1 - (i-t)^p)^{(1/p)} \\
= \; & (2^p \cdot (1 - (i - t/2)^p))^{(1/p)} - (1 - i^p)^{(1/p)} - (1 - (i-t)^p)^{(1/p)} \\
= \; & 2 \cdot (1 - (i - t/2)^p)^{(1/p)} - (1 - i^p)^{(1/p)} - (1 - (i-t)^p)^{(1/p)} \qquad (11)
\end{aligned}
$$

In order to compute the sign of (11) the function $f(x) = (1 - x^p)^{(1/p)}$ is regarded. Computing the first and second derivative of $f(x)$ yields

$$
\begin{aligned}
f'(x) &= -p \cdot x^{p-1} \cdot (1/p) \cdot (1 - x^p)^{1/p-1} = -x^{p-1} \cdot (1 - x^p)^{1/p-1} \\
f''(x) &= -(p-1) \cdot x^{p-2} \cdot (1 - x^p)^{1/p-1} + \Big(- x^{p-1} \cdot p \cdot (-x^{p-1}) \cdot (1/p - 1) \cdot (1 - x^p)^{1/p-2} \Big) \\
&= (1-p) \cdot x^{p-2} \cdot (1 - x^p)^{1/p-1} + \Big(x^{2p-2} \cdot (1-p) \cdot (1 - x^p)^{1/p-2} \Big) \\
&= (1-p) \cdot x^{p-2} \cdot (1 - x^p)^{1/p-2} \cdot \Big((1 - x^p) + x^p \Big) \\
&= (1-p) \cdot x^{p-2} \cdot (1 - x^p)^{1/p-2} \qquad (12)
\end{aligned}
$$

For $0 < x < 1$ and $p > 0$ it holds that $x^{p-2} > 0$ and $(1 - x^p)^{1/p-2} > 0$. Therefore, the sign of (12) only depends on p for $0 < x < 1$, which yields $f''(x) > 0$ for $0 < p < 1$ and $f''(x) < 0$ for $p > 1$.

So f is a strict concave function for $p > 1$, which means that the following inequality holds: $f((x_1 + x_2)/2) > f(x_1)/2 + f(x_2)/2$. Therefore, the sign of (11) is positive for $p > 1$ and $t \neq 0$, which means that $l_2(k) > l_1(k) \geq l$. It follows that $d(u, w) \leq d(u, w_1) < d(u, w_2) = 2 = d(u, v)$, which implies that (5) holds for any distance function induced by a p-norm for $1 < p < \infty$. Consequently $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$. □

(b) From the proof of Theorem 3 it follows that $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$ for any distance function induced by a p-norm for $1 < p < \infty$. An example of a graph in $\mathsf{GG}^{\rightarrow}$, which is not in $\mathsf{RNG}^{\rightarrow}$ is obtained for

$u = (-i, -j)$, $v = (i, j)$ for $0 \leq i, j \leq 1$, $m = (0, 0)$,
$w = (k, l_1(k)/2 + l_2(k)/2)$ for $0 \leq k \leq 1$ and $k \neq i$.

(c) $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ holds for an arbitrary metric and therefore also for an arbitrary distance function induced by a p-norm.

(d) $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{GG}^{\rightarrow}$ and $\mathsf{GG}^{\rightarrow} \subseteq \mathsf{P}$ implies $\mathsf{RNG}^{\rightarrow} \subseteq \mathsf{P}$

The following corollary is a direct consequence from condition (4) of Theorem 2.

Corollary 1. *For every metric induced by the p norm, except $p = 1$ and $p = \infty$ the class of relative neighborhood graphs is contained in the class of plane drawings.*

3.3.4 Distances Induced by Quasi-norms

In this subsection it is investigated, if some of the containedness relations (a)-(d) derived for metric spaces, still hold in quasi-metric spaces, where the triangle inequality is not satisfied anymore. The distance functions considered in this subsection are distance functions obtained by a quasi norm, where $0 < p < 1$ and a distance function using a relaxed version of the triangle inequality and the remaining axioms of a metric. Since the triangle inequality does not hold anymore, all containedness relations (a)-(d) have to be investigated. However, some of the containedness relations can be obtained quite easily from parts of the proof of Theorem 3. The details for testing the containedness relations (a)-(d) are presented in the following, first for a distance function induced by a p-quasinorm for $0 < p < 1$, afterwards for a distance function using a relaxed version of the triangle inequality and the remaining axioms of a metric.

Case 6: d is a distance function induced by a p-quasinorm for $0 < p < 1$.
(a) The same computation as in the proof of Theorem 3 can be used. Again $m(u, v) = (0, 0)$ and the notation $i = x(v) = -x(u)$, $j = y(v) = -y(u)$, $k = x(w)$ and $l = y(w)$ is used, assuming that $k \geq 0$ and $l \geq 0$. Also the points $w_1 = (k, l_1)$ and $w_2 = (k, l_2)$ are again chosen such that $d(m(u, v), w_1) = 1$ and $d(u, w_2) = 2$.

Since the function $f(x) = (1-x^p)^{1/p-1}$ is convex for $0 < x < 1$ it holds that $f((x_1+x_2)/2) < f(x_1)/2+f(x_2)/2$. Therefore, the sign of (11) is negative for $0 < p < 1$ and $t \neq 0$. Consequently $l_1(k) > l_2(k)$, which means that $d(m(u, v), w) < d(m(u, v), w_1) = 1$ and $2 = d(u, w_2) < d(u, w)$ for $w = (k, l_1(k)/2 + l_2(k)/2)$. Thus, $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{GG}^{\rightarrow}$.

(b) A graph in $\mathsf{GG}^{\rightarrow}$, which is not in $\mathsf{RNG}^{\rightarrow}$ is constructed, in order to prove $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$. Let $u = (-1/2, -j)$, $v = (1/2, j)$, $m(u, v) = (0, 0)$ and $w = (1, -j)$, where $j = (1-(1/2)^p)^{(1/p)}$. Then, $d(m(u, v), v) = ((1/2)^p+j^p)^{(1/p)} = ((1/2)^p+1-(1/2)^p)^{(1/p)} = 1^{(1/p)} = 1$ and consequently $d(u, v) = 2$. Furthermore, it holds that $d(u, w) = 1 + 1/2 < 2 = d(u, v)$ and $d(v, w) = ((2j)^p + (1/2)^p)^{(1/p)} = (2^p \cdot (1 - (1/2)^p) + (1/2)^p)^{(1/p)} = (2^p - 1 + (1/2)^p)^{(1/p)} < (2^p)^{(1/p)} = 2 = d(u, v)$. Therefore, the edge uv does not exist in the relative neighborhood graph.

Next the distance $d(m(u, v), w)$ is regarded. It holds that $d(m(u, v), w) = (j^p + 1^p)^{(1/p)} > 1^{(1/p)} = 1 = d(m(u, v), v)$. Therefore, the edge uv exists in the Gabriel graph. This yields $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{RNG}^{\rightarrow}$

(c) For distance functions, where the triangle inequality does not hold anymore, a Gabriel graph is not necessarily a plane drawing.

Theorem 4. *For a distance function induced by a quasi-norm with $0 < p < 1$, the class of Gabriel Graphs is not contained in the class of plane drawings.*

Proof. In order to construct a counterexample of a graph in GG, which is not in P, the following graph is considered. Let $u = (2, 0)$, $v = (-2, 0)$ and consequently $m(u, v) = (0, 0)$. An edge between u and v exists if for all vertices w, where $w \neq u$ and $w \neq v$, $d(m(u, v), w) > d(m(u, v), u)$ holds. Let $w_1 = (k, l)$ and let $w_2 = (k, -l)$ for $k \geq 0$ and $l \geq 0$. Since $d(m(u, v), u) = 2$ applying the same computation as for equation (7) yields that $d(m(u, v), w_1) = d(m(u, v), w_2) > 2$ if

$$l > (2^p - k^p)^{1/p} \tag{13}$$

Since $0 < p < 1$ it holds that $1 < 2^p < 2$. Therefore, for $k = 1$ it holds that $0 < (2^p - k^p) < 1$ and consequently also $0 < (2^p - k^p)^{1/p} < 1$. So for $l = (1/2) \cdot (1 + (2^p - 1)^{1/p})$ condition (13) is satisfied. This means that the vertices $w_1 = (1, l)$ and $w_2 = (1, -l)$ do not contradict the existence of the edge uv.

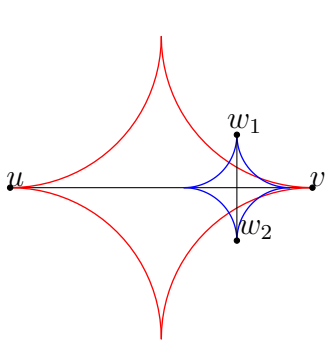
Next it is analyzed if also the edge w_1w_2 exists. The middle point of w_1 and w_2 is (1,0). Since $d(m(w_1, w_2), w_1) = l < 1 = d(m(w_1, w_2), u)$ and $d(m(w_1, w_2), w_1) = l < 3 = d(m(w_1, w_2), v)$, the vertices u and v do not contradict the existence of the edge w_1w_2. Therefore, the edges uv and w_1w_2 both exist and intersect with intersection point $q = (1, 0)$ (see Fig. 6(**a**)). Consequently GG $\not\subseteq$ P for a quasi-norm with $0 < p < 1$. □

(d) For distance functions, where the triangle inequality does not hold anymore, a relative neighboorhood graph is not necessarily a plane drawing.

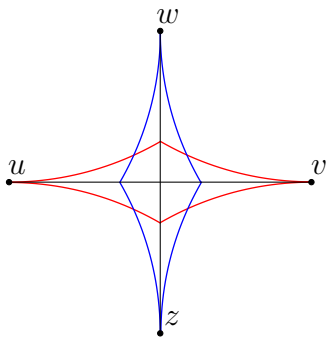
Theorem 5. *For a distance function induced by a quasi-norm, the class of relative neighborhood graphs is not contained in the class of plane drawings.*

Proof. In order to construct a counterexample of a graph in RNG, which is not in P, the following graph with $u = (-1, 0)$, $v = (1, 0)$, $w = (0, -1)$ and $z = (0, 1)$ is considered. For any distance induced by a p-quasi norm with $0 < p < 1$ the distances between two of the vertices u, v, x, z are regarded. It holds that $d(u, v) = d(w, z) = 2$ and $d(u, w) = d(v, w) = d(u, z) = d(v, z) = (1^p + 1^p)^{(1/p)} = 2^{(1/p)} > 2$. Therefore, both edges uv and wz exists and intersect in the point $q = (0, 0)$. This example shows that RNG $\not\subseteq$ P (see Fig. 6(**b**)) □

For testing GG $\subseteq$ P and RNG $\subseteq$ P, H-PILoT did not terminate after one hour using the encoding for the concrete quasi-norm of $p = 1/2$, when using the same formalization as in the test for the case of $p = 2$. Therefore again, manual proof were required.



(**a**) The graph constructed in the proof of Theorem 4



(**b**) The graph constructed in the proof of Theorem 5

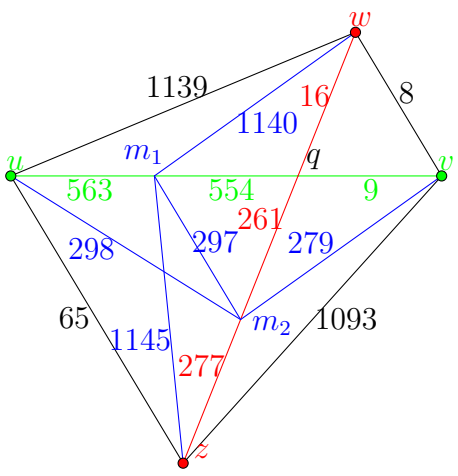
Figure 6: Illustration of the example graphs proving that $\mathsf{GG} \not\subseteq \mathsf{P}$ and $\mathsf{RNG} \not\subseteq \mathsf{P}$ for distance function induced by a p-quasinorm for $0 < p < 1$

Case 7: d is a distance function defined by a relaxed triangle inequality

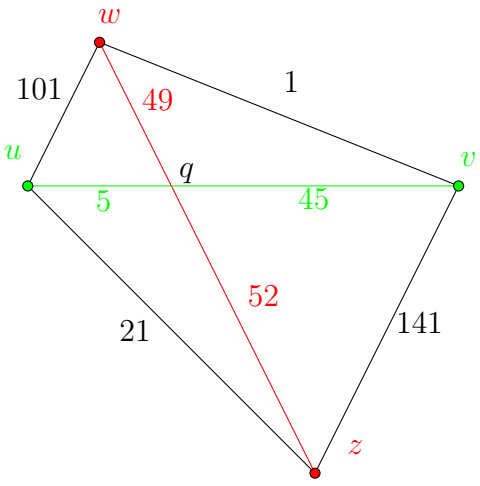
Since none of the containedness relations (a)-(d) holds for a distance function induced by a p-quasinorm for $0 < p < 1$, also none of the containedness relations holds for a distance function defined by a relaxed triangle inequality and the remaining axioms of a metric. It is therefore only investigated if some of the counterexamples can be obtained by HPILoT. In the following, the relaxed triangle inequality

$$\forall u, v, w : d(u, w) \leq 2 \cdot (d(u, v) + d(v, w)) \tag{14}$$

is used together with the remaining axioms of a metric to define a distance function. The following graphs illustrated in Fig 7 are returned by H-PILoT, for proving $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{P}$ (see Fig. 7(**a**)) and $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{P}$ (see Fig. 7(**b**)).



(**a**) An example proving $\mathsf{GG}^{\rightarrow} \not\subseteq \mathsf{P}$



(**b**) An example proving $\mathsf{RNG}^{\rightarrow} \not\subseteq \mathsf{P}$

Figure 7: Counterexamples for a distance function with relaxed triangle inequality

When the factor 2 in equation (14) was replaced by another rational number $K > 1$, H-PILoT returned similar counterexamples. The test for creating the counterexamples can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/Section3.

3.3.5 Overview about the containedness relations for all considered distance functions

The following table shows which of the containedness relations hold for the according distance function. Moreover, it is listed whether a proof or model for the according containedness relation was derived manually (m), by H-PiLoT (H) or as a consequence of other containedness relations (c).

Distance Function	RNG $\subseteq$ GG	GG $\subseteq$ RNG	GG $\subseteq$ P	RNG $\subseteq$ P
Euclidean distance ($p = 2$)	✓ (H)	✗ (H)	✓ (c)	✓ (c)
Arbitrary metric	✗ (H)	✗ (H)	✓ (H)	✗ (c)
Manhattan metric ($p = 1$)	✗ (H)	✗ (H)	✓ (c)	✗ (H)
Maximum metric ($p = \infty$)	✗ (H)	✗ (H)	✓ (c)	✗ (H)
p-norm with $1 < p < \infty$	✓ (m)	✗ (m)	✓ (c)	✓ (c)
Quasi-norm with $0 < p < 1$	✗ (m)	✗ (m)	✗ (m)	✗ (m)
Relaxed triangle inequality	✗ (c)	✗ (c)	✗ (H)	✗ (H)

Table 2: Overview about the containedness relations depending on the distance function.

4 Graph Class Containedness Relations Between Graph Classes Obtained by Simple Transformations

In contrast to the graph classes regarded in the previous section, a graph class can also be defined in a way, such that for a given vertex set multiple graphs exist, that satisfy the properties defining the graph class. An example for such a graph class, is the class of Quasi-Unit-Disk graphs, where some edges have to exist, some edges cannot exist and some edges can exist, but do not have to exist. The conditions for the edges that have to exist are expressed by an inclusion predicate, while the condition for edges that cannot exist are expressed by an exclusion predicate. Another type of predicate that can be considered, is a transfer predicate. A transfer predicate is used to express a condition, when the existence of one edge enforces the existence of another edge. Graph classes are expressed by axioms for the existence resp. non-existence of edges using the inclusion, exclusion and transfer predicate.

The graphs considered in this section are directed, i.e. each edge is assigned with an orientation. The investigated graph classes are obtained by applying simple transformations on graph classes defined by inclusion, exclusion and transfer axioms. A simple transformation maps a source graph to an image graph, where the existence of an edge in the image graph only depends on the existence of this edge and its reversed edge in the source graph. Two examples for simple transformations are the supergraph and subgraph transformation, where a symetric graph is obtained by adding missing reversed edges resp. deleting edges where the reversed edge is missing. For such graph classes obtained after applying simple transformations, it is investigated, when one class is contained in another graph class, i.e all graphs of one graph class are also contained in the other class.
Parts of the results in this chapter were already published in [29] and [12].

4.1 Axiomatic Description of Graph Classes

In this section graph classes C are considered, where each graph $G = (V, E)$ consists of vertices V of a vertex universe Ω and an edge relation E. The edge relation uses a binary predicate E, where $E(u, v)$ is true iff $uv \in E$. Furthermore, the edge relation satisfies the axioms

$$\forall u, v \in V : \pi^i_{\mathsf{C}}(u, v) \Rightarrow E(u, v) \tag{15}$$

$$\forall u, v \in V : \pi^e_{\mathsf{C}}(u, v) \Rightarrow \neg E(u, v) \tag{16}$$

$$\forall u, v, w \in V : E(u, w) \wedge \pi^t_{\mathsf{C}}(u, v, w) \Rightarrow E(u, v) \tag{17}$$

$$\forall u, v, w \in V : \neg E(u, w) \wedge \pi^t_{\mathsf{C}}(u, w, v) \Rightarrow \neg E(u, v) \tag{18}$$

The predicates $\pi^i_{\mathsf{C}}(u, v)$, $\pi^e_{\mathsf{C}}(u, v)$ and $\pi^t_{\mathsf{C}}(u, v, w)$ are termed as *inclusion*, *exclusion*, and *transfer predicates*, respectively. It is referred to the conditions (15)–(17) as *inclusion*, *exclusion* and *transfer axioms*, accordingly (though conditions (17) and (18) are equivalent, both are

listed here for later reference).

The graphs in this section are directed, i.e. the edge relation E is considered separately for the vertex pair (u, v) and the vertex pair (v, u). Furthermore, the graphs do not contain self-loops, i.e. an edge from a vertex to itself. In order to avoid the existence of self-loops also in the complement graph, the edge relation is only defined for vertex pairs between different vertices. Furthermore, the inclusion, exclusion and transfer predicate regarded in this section cannot be true if a pair of equal vertices is considered. This means, that for all $u \in V$ the predicates $\pi^i_{\mathsf{C}}(u, u)$ and $\pi^e_{\mathsf{C}}(u, u)$ are false and for all $u, v \in V$ the predicates $\pi^t_{\mathsf{C}}(u, u, v)$, $\pi^t_{\mathsf{C}}(u, v, u)$ and $\pi^t_{\mathsf{C}}(u, v, v)$ are false.

A graph $G = (V, E)$ is in class C (expressed as $G \in \mathsf{C}$) if G satisfies the conditions (15)–(17). In this case, also sometimes G *satisfies class* C is written. For a graph class C that consists of all graphs satisfying a set of axioms of the form (15)–(17), it is referred to C as a graph class defined by inclusion, exclusion, and transfer axioms. In this section, such graph classes and transformations on them are studied. For this the following additional assumptions are made:

Assumption 1. (Independence) *The inclusion, exclusion and transfer predicates are determined by the properties of their arguments $u, v, w \in V$ and are independent of any other vertices in V. This can be ensured, for instance, by requiring that these predicates are expressed as quantifier-free formulae over a background theory.*

The inclusion, exclusion, and transfer predicates used in the axioms for the classes $\mathsf{MinDG}(\mathsf{r})$, $\mathsf{MaxDG}(\mathsf{r})$ and CRG can be expressed as quantifier-free formulae over a theory in which one can talk about points and distances between points:

$$\begin{aligned} \pi^i_{\mathsf{C}}(u, v) &:= u \neq v \wedge d(u, v) \leq r \\ \pi^e_{\mathsf{C}}(u, v) &:= d(u, v) > r \\ \pi^t_{\mathsf{C}}(u, v, w) &:= u \neq v \wedge d(u, v) \leq d(u, w) \end{aligned}$$

Assumption 2. (Transitivity) *The transfer predicates $\pi^t_{\mathsf{C}}(u, v, w)$ is* transitive*, i.e.,*

$$\pi^t_{\mathsf{C}}(u, v, w) \wedge \pi^t_{\mathsf{C}}(u, w, x) \Rightarrow \pi^t_{\mathsf{C}}(u, v, x)$$

Obviously, the transfer predicate used in the axiom for CRG is transitive.

Assumption 3. (Soundness) *The inclusion, exclusion, and transitive transfer predicates are* sound*, which means that they are not contradicting:*

$$\begin{aligned} \forall u, v \in V : \pi^i_{\mathsf{C}}(u, v) &\Rightarrow \neg \pi^e_{\mathsf{C}}(u, v) \\ \forall u, v, w \in V : \pi^i_{\mathsf{C}}(u, w) \wedge \pi^t_{\mathsf{C}}(u, v, w) &\Rightarrow \neg \pi^e_{\mathsf{C}}(u, v) \end{aligned}$$

A graph class is termed to be sound in this case.

It is easy to see, as long as $r_1 \leq r_2$ any conjunction of the predicates used in the axioms for $\mathsf{MinDG}(r_1)$, $\mathsf{MaxDG}(r_2)$ and CRG satisfies the soundness condition.

4.2 Simple Graph Class Transformations

The most general form of a graph transformation γ is a mapping which transforms a given graph G into a graph H. A graph transformation applied on a graph class C results in a graph class $\gamma(\mathsf{C})$, which consists of all graphs H, for which there exists a graph $G \in \mathsf{C}$ with $H = \gamma(G)$.

A graph transformation γ is *vertex preserving* if $\gamma(G)$ has the same vertex set as G, i.e., given $G = (V, E)$, the transformed graph is $\gamma(G) = (V, F)$ for some F. Obviously, a vertex preserving transformation γ on the *null graph* $(\emptyset, \emptyset)$ yields the null graph.

A vertex preserving graph transformation is *simple* if there exists a Boolean function in two variables $\delta : \mathbb{B}^2 \to \mathbb{B}$ (where $\mathbb{B}$ is the two-element Boolean algebra) such that for all graphs $G = (V, E)$ the transformation $\gamma(G) = (V, F)$ satisfies

$$\forall u, v \in V, u \neq v : F(u, v) = \delta(E(u, v), E(v, u))$$

The symmetric super graph $(\cdot)^+$ and symmetric subgraph $(\cdot)^-$ are obviously simple transformations with boolean functions $\vee$ and $\wedge$, respectively.

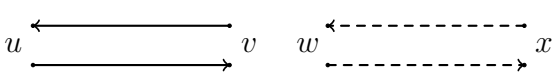
As there are in total $4^2 = 16$ possible Boolean functions of the form $\delta : \mathbb{B}^2 \to \mathbb{B}$, there exist 16 possible vertex preserving simple graph transformations, as shown in the following tables and Fig. 8

		$\delta(E(u,v), E(v,u))$					
		$E(u,v)$	0	0	1	1	
Name	Symbol	$E(v,u)$	0	1	0	1	Property
Symmetric subgraph	G^-	$F(u,v)$	0	0	0	1	Symmetric
Symmetric supergraph	G^+	$F(u,v)$	0	1	1	1	Symmetric
XOR graph	$G^\oplus$	$F(u,v)$	0	1	1	0	Symmetric
Empty graph	$\mathsf{G}^\perp$	$F(u,v)$	0	0	0	0	Symmetric
Identity graph	G^{id}	$F(u,v)$	0	0	1	1	
Reversed graph	$G^\leftrightarrow$	$F(u,v)$	0	1	0	1	
Outgoing directed subgraph	$G^\rightarrow$	$F(u,v)$	0	0	1	0	Oriented
Incoming directed subgraph	$G^\leftarrow$	$F(u,v)$	0	1	0	0	Oriented

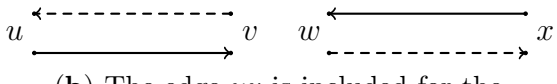
Table 3: The eight non-negated vertex preserving simple graph transformations.

	$\delta(E(u,v),E(v,u))$				
	$E(u,v)$	0	0	1	1
Symbol	$E(v,u)$	0	1	0	1	Property
$G^{\neg-}$	$F(u,v)$	1	1	1	0	Symmetric
$G^{\neg+}$	$F(u,v)$	1	0	0	0	Symmetric
$G^{\neg\oplus}$	$F(u,v)$	1	0	0	1	Symmetric
$G^{\neg\perp}$	$F(u,v)$	1	1	1	1	Symmetric
$G^{\neg\mathrm{id}}$	$F(u,v)$	1	1	0	0	
$G^{\neg\leftrightarrow}$	$F(u,v)$	1	0	1	0	
$G^{\neg\rightarrow}$	$F(u,v)$	1	1	0	1	Complementary oriented
$G^{\neg\leftarrow}$	$F(u,v)$	1	0	1	1	Complementary oriented

Table 4: The eight negated vertex preserving simple graph transformations.



(**a**) The edge uv is included for the non-negated transformations $+$, $-$, id, $\leftrightarrow$, the edge wx only for the negated transformations



(**b**) The edge uv is included for the non-negated transformations $+$, $\oplus$, id, $\rightarrow$, the edge wx for the non-negated transformations $+$, $\oplus$, $\leftrightarrow$, $\leftarrow$

Figure 8: Illustration of the simple vertex transformations

In both tables, all four possible combinations of existence or non-existence of the edges uv and vu in the original graph are shown in the table header by the column values 0 (non-existence) and 1 (existence) for $E(u,v)$ and $E(v,u)$. For each of those possible cases, the edge uv can exist, i.e. $F(u,v)=1$, or not exist, i.e. $F(u,v)=0$, in the resulting image graph. All possible combinations are shown by the eight rows of both tables (i.e., 16 in total) by the values 0 and 1 for $F(u,v)$.

For each transformation a symbol and a mnemonic in Table 3 is defined. For each transformation in that table, the negated form can be defined, as listed in Table 4. Given a transformation γ, the negated transformation $\neg\gamma$ yields the complement graph $\overline{G}=(V,\overline{F})$ of the graph $G=(V,F)$ resulting from γ, i.e., each existing edge in G does not exist in $\overline{G}$ and each edge not existing in G exists $\overline{G}$. Therefore, the edge sets F and $\overline{F}$ are disjoint. Furthermore, $F\cup\overline{F}$ corresponds to the set of all vertex pairs in $V\times V$, except the vertex pairs of equal vertices. Specifically for the transformation empty graph $G^{\perp}$, the negated empty graph $G^{\neg\perp}$ is the complete graph, of course.

In the last column of Table 3 it is listed whether the transformations always yield a *symmetric* graph (i.e. $\forall u,v\in V: F(u,v)\Rightarrow F(v,u)$), always yield an *oriented* graph (i.e.

$\forall u, v \in V : F(u,v) \Rightarrow \neg F(v,u))$, or the image graph is neither necessarily symmetric nor oriented (column entry is kept empty). As shown in Table 4, the symmetry is preserved for the complementary graphs given by the negated transformations. However, the last two shown transformations yield graphs, termed as *complementary oriented* (i.e. $\forall u, v \in V : \neg F(u,v) \Rightarrow F(v,u)$), which means that the complement of the graph is oriented.

In the following discussion, it has to be distinguished between the transformations for XOR $G^{\oplus}$, outgoing $G^{\rightarrow}$ and incoming $G^{\leftarrow}$ on a graph G and as well the corresponding negated transformations for XOR $G^{\neg\oplus}$, outgoing $G^{\neg\rightarrow}$ and incoming $G^{\neg\leftarrow}$ on the one hand and all remaining simple transformations on the other hand. It turns out that for the remaining 10 transformations a small counter example, with at most four vertices can be found if a containedness relation does not hold. For each of the two XOR, incoming and outgoing transformations, this is not the case. For these six transformations, arbitrary long chains of edge dependencies to determine minimum counter examples are possible. The proofs are given in Lemma 7 for the outgoing and incoming transformations and in Lemma 11 for the XOR transformations.

To distinguish between those situations where all transformation are considered and where only a subset of transformations is considered the following sets of transformations are defined

$$\Gamma_1 = \{(\cdot)^x, (\cdot)^{\neg x} : x = -, +, \perp, \mathrm{id}, \leftrightarrow\},$$

$$\Gamma_2 = \{(\cdot)^x, (\cdot)^{\neg x} : x = \rightarrow, \leftarrow\},$$

$$\Gamma_3 = \{(\cdot)^x, (\cdot)^{\neg x} : x = \oplus\}.$$

For illustrations in the following subsections the color encoding form Fig.9 is used.

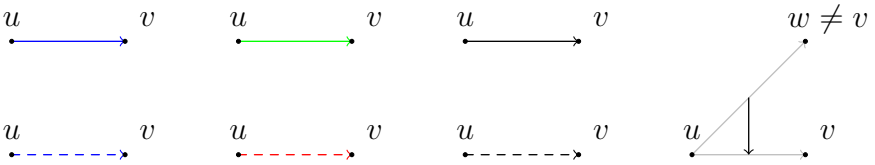


Figure 9: Illustration of the used notation. Solid blue line: $uv \in E$, dashed blue line: $uv \notin E$, solid green line: $uv \in F$, dashed red line: $uv \notin F$, solid black line: $\pi^i_{\mathsf{B}}(u,v)$, dashed black line $\pi^e_{\mathsf{B}}(u,v)$, and black link from uw to uv: $\pi^t_{\mathsf{B}}(u,v,w)$ holds.

4.3 Proving Graph Class Containedness Relations

4.3.1 General Properties

In the following, some general properties for class containedness relations of graph classes described by axioms using inclusion, exclusion and transfer predicates, which satisfy Independence (Assumption 1), and graph classes obtained by applying a simple transformation on

such graph classes are analyzed. It is proven that such graph classes are closed under induced subgraphs (according to Def. 20). Moreover, given a graph class described by axioms using sound predicates, for each finite vertex set V, an edge configuration can be found such that the given graph class axioms are satisfied.

Lemma 2. *Let C be a graph class described by axioms using inclusion, exclusion and transfer predicates, satisfying Independency, Transitivity and Soundness (Assumptions 1–3). If $G = (V, E)$ satisfies C, then every induced subgraph G' with $V' \subseteq V$ satisfies C.*

Proof. Consider a graph $G = (V, E)$, which satisfies C. Axioms (15)–(17) are satisfied for V. Let $V' \subseteq G$. Consider one of the conditions defining the class C, say condition (15):

$$\forall u, v : \ \pi^i_{\mathsf{C}}(u, v) \Rightarrow E(u, v).$$

Let $u, v \in V'$. Then, $u, v \in V$. Assume that $\pi^i_{\mathsf{C}}(u, v)$ is true (relative to the set of edges, V'). By Independence (Assumption 1), the value of this predicate is independent of the other vertices in the graph, so $\pi^i_{\mathsf{C}}(u, v)$ is true also when the whole set of vertices in $G = (V, E)$ is considered. As G satisfies condition (15), it follows that $E(u, v)$ is true (i.e., there exists an edge from u to v in G). Since $E' = E \cap (V' \times V')$, there also exists an edge from u to v in G'.

The same line of argumentation can be used for conditions (16) and (17). This shows that also the subgraph G' induced by V' satisfies C. □

Remark 3. *Since the null graph is always sound and every graph class contains at least the null graph, Lemma 2 is vacuously true if Soundness (Assumption 3) does not hold.*

If Independence (Assumption 1) is not satisfied, then Lemma 2 does not necessarily hold. Consider, for instance, the graph class C described by axioms of the form (15)–(17), where for all u, v, x:

$$\pi^i_{\mathsf{C}}(u, v) = \forall w((w \neq u \wedge w \neq v) \Rightarrow (E(u, w) \wedge E(w, v)))$$
$$\pi^e_{\mathsf{C}}(u, v) = \text{false}, \quad \pi^t_{\mathsf{C}}(u, x, v) = \text{false}.$$

In this graph class, the graph $G = (\{a, b, c, d, e\}, \{(a, b), (b, e), (a, d), (d, e)\})$ satisfies the axioms (15)–(17), but the subgraph induced by $V' = \{a, b, d, e\}$ does not satisfy axiom (15).

Lemma 3. *Let C be a graph class defined by axioms using inclusion, exclusion and transfer predicates satisfying Independency, Transitivity and Soundness (Assumptions 1–3) and γ be a vertex preserving graph transformation. If γ is simple and if $H = (V, F)$ satisfies $\gamma(\mathsf{C})$, then every subgraph H' induced by $V' \subseteq V$ satisfies $\gamma(\mathsf{C})$.*

Proof. Consider a graph $H = (V, F)$, which satisfies $\gamma(\mathsf{C})$. With γ being vertex preserving, there exists a graph $G = (V, E) \in \mathsf{C}$ such that $H = \gamma(G)$. Let $V' \subseteq V$, $H' = (V', F')$ be the subgraph of H and $G' = (V', E')$ be the subgraph of G induced by V', respectively.

Let δ be the boolean function of the simple transformation γ. For all $u, v \in V'$ holds: uv is an edge of $\gamma(G')$ iff

$$\begin{aligned}\delta(E'(u,v), E'(v,u)) &\Leftrightarrow \delta(E(u,v), E(v,u)) \\ &\Leftrightarrow F(u,v) \Leftrightarrow F'(u,v) \end{aligned} \tag{19}$$

This implies $H' = \gamma(G')$, and thus $H' \in \gamma(\mathsf{C})$, since $G' \in \mathsf{C}$ with Lemma 2. □

Remark 4. *The second part of Lemma 3 does not necessarily hold if the transformation γ is not simple. Consider, for instance, the transformation defined by*

$$\forall u, v\Big(\ \phi(u,v) \rightarrow F(u,v)\Big),$$
$$\forall u, v\Big(\neg\phi(u,v) \rightarrow \neg F(u,v)\Big)$$

where $\phi(u,v) := \forall w\Big((u \neq w \wedge v \neq w) \rightarrow (E(u,w) \wedge E(w,v))\Big)$.

Let C be a graph class defined by inclusion, exclusion and transfer axioms containing the graph.
$G = (\{a,b,c,d\}, \{(a,b),(b,d)\})$ *due to the fact that in this specific graph* $\pi^i_{\mathsf{C}}(a,b) = \pi^i_{\mathsf{C}}(b,d) =$ true, $\pi^e_{\mathsf{C}}(x,y) =$ true *for all other pairs of vertices, and* $\pi^t_{\mathsf{C}}(u,v,w)$ *is false everywhere. Then* $H = \gamma(G) = (\{a,b,c,d\}, \emptyset) \in \gamma(\mathsf{C})$. *However, the subgraph H' of H induced by $V' = \{a,b,d\}$ is not in $\gamma(\mathsf{C})$: Assume there exists a graph $G' \in \mathsf{C}$ with $H' = \gamma(G')$. Since $G' \in \mathsf{C}$, due to the fact that* $\pi^i_{\mathsf{C}}(a,b) = \pi^i_{\mathsf{C}}(b,d) =$ true, $\pi^e_{\mathsf{C}}(x,y) =$ true *for all other pairs of vertices, and* π^t_{C} *is false everywhere, the set of edges of G' must be $E = \{(a,b),(b,d)\}$. But then in this case $\forall w(a \neq w \wedge d \neq w \rightarrow E(a,w) \wedge E(w,d))$ holds, so in $\gamma(G')$, an edge (a,d) should exist. However, this edge does not exist in H'.*

Lemma 4. *Let C be a graph class defined over axioms using predicates satisfying Assumptions 1-3 (Independence, Soundness, Transitivity). For each finite set $V \subseteq \Omega$ at least one graph $G = (V, E)$ exists, which satisfies C.*

Proof. Consider any such set V. Set $E(u,v) =$ true if $\pi^i_{\mathsf{C}}(u,v)$ holds or if there exist a $w \in V$ such that $\pi^i_{\mathsf{C}}(u,w) \wedge \pi^t_{\mathsf{C}}(u,v,w)$ holds. Otherwise set $E(u,v) =$ false. This configuration obviously satisfies (15) and (17), and does not contradict (16) since the conditions are sound. Thus, the such defined graph $G = (V, E)$ satisfies C. □

Note that if C is a graph class defined by inclusion, exclusion and transfer axioms and γ is a simple vertex preserving transformation, it is not immediate that $\gamma(\mathsf{C})$ can also be described by inclusion, exclusion and transfer axioms.
In Section 4.3.2, it is shown that when $\gamma \in \Gamma_1$, then the properties of the edge-relation F of graphs $H = (V, F)$ in $\gamma(\mathsf{C})$ can be described using axioms in which only F and the inclusion, exclusion and transfer predicates $\pi^i_{\mathsf{C}}(u,v), \pi^e_{\mathsf{C}}(u,v)$ and $\pi^t_{\mathsf{C}}(u,v,w)$ are used.

4.3.2 Properties for the Graph Classes Obtained by Transformations in Γ_1

Next the connection between edge configurations $F(u, v)$ for a graph being in a class obtained by a transformation in Γ_1 and the inclusion, exclusion and transfer predicates of that class is considered. In the following, the notation B^γ is used instead of $\gamma(\mathsf{B})$ in order to use the notation for graph transformations defined in Table 3 and Table 4.

Lemma 5. *Let B be a sound graph class described by axioms using inclusion, exclusion and transitive transfer predicates $\pi^i_{\mathsf{B}}(u, v)$, $\pi^e_{\mathsf{B}}(u, v)$, $\pi^t_{\mathsf{B}}(u, v, w)$, respectively. Let γ be any of the vertex preserving, simple graph transformations in Γ_1. A graph $H = (V, F)$ is in class B^γ, iff for every $u, v \in V$ the edge relation $F(u, v)$ satisfies the implications depicted in Table 5 for that transformation.*

Proof. Two implication directions "⇒" and "⇐" have to be shown.

Part 1 (⇒). With $H = (V, F) \in \mathsf{B}^\gamma$ there exists a graph $G = (V, E) \in \mathsf{B}$ with $H = G^\gamma$. The proof is split up in two subparts. The first part contains the proof that holds for all transformations in Γ_1 and in the second part, transformation specific proofs are considered.

Subpart 1 (all transformations in Γ_1)**.** If an edge is required to exist in G, i.e., $E(u, v)$, then the contraposition of graph class axiom (16) yields

$$E(u, v) \Rightarrow \neg\pi^e_{\mathsf{B}}(u, v) \tag{20}$$

Moreover, the contraposition of graph class axiom (18) yields

$$E(u, v) \Rightarrow \forall x : E(u, x) \vee \neg\pi^t_{\mathsf{B}}(u, x, v) \tag{21}$$

Since the transfer predicate is transitive, it is not required to use (21) again for $E(u, x)$.

If an edge is forbidden to exist in G, i.e. $\neg E(u, v)$ then the contraposition of graph class axiom (15) yields

$$\neg E(u, v) \Rightarrow \neg\pi^i_{\mathsf{B}}(u, v) \tag{22}$$

Moreover, the contraposition of graph class axiom (17) yields

$$\neg E(u, v) \Rightarrow \forall x : \neg E(u, x) \vee \neg\pi^t_{\mathsf{B}}(u, v, x) \tag{23}$$

Also for $\neg E(u, x)$ it is not required to use (23) again, since the transfer predicate is transitive.

γ	**Edge Inclusion and Exclusion Implications**			$\mu(\gamma)$
$H \in \mathsf{B}^{\bot}$	$F(u,v)$	$\Rightarrow$	$\bot$	2
	$\neg F(u,v)$	$\Rightarrow$	$\top$	
$H \in \mathsf{B}^{\neg\bot}$	$F(u,v)$	$\Rightarrow$	$\top$	2
	$\neg F(u,v)$	$\Rightarrow$	$\bot$	
$H \in \mathsf{B}$	$F(u,v)$	$\Rightarrow$	$\neg\pi^e_{\mathsf{B}}(u,v)$	3
	$\neg F(u,v)$	$\Rightarrow$	$\neg\pi^i_{\mathsf{B}}(u,v) \wedge \forall w : \Big(\neg F(u,w) \vee \neg\pi^t_{\mathsf{B}}(u,v,w)\Big)$	
$H \in \mathsf{B}^{\neg}$	$F(u,v)$	$\Rightarrow$	$\neg\pi^i_{\mathsf{B}}(u,v)$	3
	$\neg F(u,v)$	$\Rightarrow$	$\neg\pi^e_{\mathsf{B}}(u,v) \wedge \forall w : \Big(\neg F(u,w) \vee \neg\pi^t_{\mathsf{B}}(u,v,w)\Big)$	
$H \in \mathsf{B}^{\leftrightarrow}$	$F(u,v)$	$\Rightarrow$	$\neg\pi^e_{\mathsf{B}}(v,u)$	3
	$\neg F(u,v)$	$\Rightarrow$	$\neg\pi^i_{\mathsf{B}}(v,u) \wedge \forall w : \Big(\neg F(w,v) \vee \neg\pi^t_{\mathsf{B}}(v,u,w)\Big)$	
$H \in \mathsf{B}^{\neg\leftrightarrow}$	$F(u,v)$	$\Rightarrow$	$\neg\pi^i_{\mathsf{B}}(v,u) \wedge \forall w : \Big(F(w,v) \vee \neg\pi^t_{\mathsf{B}}(v,u,w)\Big)$	3
	$\neg F(u,v)$	$\Rightarrow$	$\neg\pi^e_{\mathsf{B}}(v,u)$	
$H \in \mathsf{B}^{-}$	$F(u,v)$	$\Rightarrow$	$F(v,u) \wedge \neg\pi^e_{\mathsf{B}}(u,v) \wedge \forall w : \Big(\neg\pi^e_{\mathsf{B}}(u,w) \vee \neg\pi^t_{\mathsf{B}}(u,w,v)\Big)$	4
	$\neg F(u,v)$	$\Rightarrow$	$\Big(\neg\pi^i_{\mathsf{B}}(u,v) \wedge \forall x : \Big(\Big(\neg F(u,x) \wedge \neg\pi^i_{\mathsf{B}}(u,x)\Big) \vee \neg\pi^t_{\mathsf{B}}(u,v,x)\Big)\Big) \vee$	
			$\Big(\neg\pi^i_{\mathsf{B}}(v,u) \wedge \forall y : \Big(\Big(\neg F(v,y) \wedge \neg\pi^i_{\mathsf{B}}(v,y)\Big) \vee \neg\pi^t_{\mathsf{B}}(v,u,y)\Big)\Big)$	
$H \in \mathsf{B}^{\neg-}$	$F(u,v)$	$\Rightarrow$	$\Big(\neg\pi^i_{\mathsf{B}}(u,v) \wedge \forall x : \Big(\Big(F(u,x) \wedge \neg\pi^i_{\mathsf{B}}(u,x)\Big) \vee \neg\pi^t_{\mathsf{B}}(u,v,x)\Big)\Big) \vee$	4
			$\Big(\neg\pi^i_{\mathsf{B}}(v,u) \wedge \forall y : \Big(\Big(F(v,y) \wedge \neg\pi^i_{\mathsf{B}}(v,y)\Big) \vee \neg\pi^t_{\mathsf{B}}(v,u,y)\Big)\Big)$	
	$\neg F(u,v)$	$\Rightarrow$	$\neg F(v,u) \wedge \neg\pi^e_{\mathsf{B}}(u,v) \wedge \forall w : \Big(\neg\pi^e_{\mathsf{B}}(u,w) \vee \neg\pi^t_{\mathsf{B}}(u,w,v)\Big)$	
$H \in \mathsf{B}^{+}$	$F(u,v)$	$\Rightarrow$	$\Big(\neg\pi^e_{\mathsf{B}}(u,v) \wedge \forall x : \Big(\Big(F(u,x) \wedge \neg\pi^e_{\mathsf{B}}(u,x)\Big) \vee \neg\pi^t_{\mathsf{B}}(u,x,v)\Big)\Big) \vee$	4
			$\Big(\neg\pi^e_{\mathsf{B}}(v,u) \wedge \forall y : \Big(\Big(F(v,y) \wedge \neg\pi^e_{\mathsf{B}}(v,y)\Big) \vee \neg\pi^t_{\mathsf{B}}(v,y,u)\Big)\Big)$	
	$\neg F(u,v)$	$\Rightarrow$	$\neg F(v,u) \wedge \neg\pi^i_{\mathsf{B}}(u,v) \wedge \forall w : \Big(\neg\pi^i_{\mathsf{B}}(u,w) \vee \neg\pi^t_{\mathsf{B}}(u,v,w)\Big)$	
$H \in \mathsf{B}^{\neg+}$	$F(u,v)$	$\Rightarrow$	$F(v,u) \wedge \neg\pi^i_{\mathsf{B}}(u,v) \wedge \forall w : \Big(\neg\pi^i_{\mathsf{B}}(u,w) \vee \neg\pi^t_{\mathsf{B}}(u,v,w)\Big)$	4
	$\neg F(u,v)$	$\Rightarrow$	$\Big(\neg\pi^e_{\mathsf{B}}(u,v) \wedge \forall x : \Big(\Big(\neg F(u,x) \wedge \neg\pi^e_{\mathsf{B}}(u,x)\Big) \vee \neg\pi^t_{\mathsf{B}}(u,x,v)\Big)\Big) \vee$	
			$\Big(\neg\pi^e_{\mathsf{B}}(v,u) \wedge \forall y : \Big(\Big(\neg F(v,y) \wedge \neg\pi^e_{\mathsf{B}}(v,y)\Big) \vee \neg\pi^t_{\mathsf{B}}(v,y,u)\Big)\Big)$	

Table 5: The expressions implied by edge existence and non-existence under the considered vertex preserving simple graph class transformations in Γ_1. Quantification is over all vertices V of the considered graph $H = (V, F)$. The maximum size of a minimum witness graph to find a counter example is expressed by $\mu(\gamma)$.

Subpart 2 (transformation specific part of the proof)**.**
Identity transformation
For the identity transformation B $F(u,v) \Leftrightarrow E(u,v)$ holds.
This yields the following implications:

$$F(u,v) \Rightarrow E(u,v) \tag{24}$$
$$\neg F(u,v) \Rightarrow \neg E(u,v) \tag{25}$$
$$E(u,v) \Rightarrow F(u,v) \tag{26}$$
$$\neg E(u,v) \Rightarrow \neg F(u,v) \tag{27}$$

From (24) with (21) the expression

$$F(u,v) \Rightarrow E(u,v) \wedge \forall w : \Big(E(u,w) \vee \neg \pi^t_{\mathsf{B}}(u,w,v)\Big) \tag{28}$$

is derived. Using (26) and (20) in (28) yields the expression in Table 5.
Moreover, from (25) with (23) the expression

$$\neg F(u,v) \Rightarrow \neg E(u,v) \wedge \forall w : \Big(\neg E(u,w) \vee \neg \pi^t_{\mathsf{B}}(u,v,w)\Big) \tag{29}$$

is derived. Using (27) and (22) in (29) yields the expression in Table 5.

Reverse transformation For the $\mathsf{B}^{\leftrightarrow}$ transformation, $F(u,v) \Leftrightarrow E(v,u)$ holds.
This yields the following implications:

$$F(u,v) \Rightarrow E(v,u) \tag{30}$$
$$\neg F(u,v) \Rightarrow \neg E(v,u) \tag{31}$$
$$E(v,u) \Rightarrow F(u,v) \tag{32}$$
$$\neg E(v,u) \Rightarrow \neg F(u,v) \tag{33}$$

From (30) with (21) the expression

$$F(u,v) \Rightarrow E(v,u) \vee \forall w : \Big(E(w,u) \vee \neg \pi^t_{\mathsf{B}}(v,w,u)\Big) \tag{34}$$

is derived. Using (32) and (20) in (34) yields the expression in Table 5.
Moreover, from (31) with (23) the expression

$$\neg F(u,v) \Rightarrow \neg E(v,u) \wedge \forall w : \Big(\neg E(v,w) \vee \neg \pi^t_{\mathsf{B}}(v,w,u)\Big) \tag{35}$$

is derived. Using (22) and (33) in (35) yields the formula in Table 5.

Supergraph transformation For the B^+ transformation $F(u,v) \Leftrightarrow E(u,v) \vee E(v,u)$ holds. This yields the following implications:

$$F(u,v) \Rightarrow E(u,v) \vee E(v,u) \tag{36}$$
$$\neg F(u,v) \Rightarrow \neg E(u,v) \wedge \neg E(v,u) \tag{37}$$
$$E(u,v) \Rightarrow F(u,v) \tag{38}$$

Furthermore, the transformation is symmetric. Therefore, one of the following implications have to be added in the formula:

$$F(u,v) \Rightarrow F(v,u) \tag{39}$$
$$\neg F(u,v) \Rightarrow \neg F(v,u) \tag{40}$$

From (36) with (21) the expression expression

$$F(u,v) \Rightarrow \left(E(u,v) \wedge \forall x : \left(E(u,x) \vee \neg \pi^t_{\mathsf{B}}(u,x,v) \right) \right) \vee \left(E(v,u) \wedge \forall y : \left(E(v,y) \vee \neg \pi^t_{\mathsf{B}}(v,y,u) \right) \right) \tag{41}$$

is derived. Using (20), (38) and (39) in (41) yields the formula in the table.
Furthermore, from (37) with (23) the expression

$$\neg F(u,v) \Rightarrow \left(\neg E(u,v) \wedge \forall x : \left(\neg E(u,x) \vee \neg \pi^t_{\mathsf{B}}(u,v,x) \right) \right) \wedge \left(\neg E(v,u) \wedge \forall y : \left(\neg E(v,y) \vee \neg \pi^t_{\mathsf{B}}(v,u,y) \right) \right) \tag{42}$$

is derived. Next (22) and (40) are used in (42).
Since the transformation is symmetric and (40) is included in the formula for $\neg F(u,v)$, it is enough to list only the condition for $\neg E(u,v)$ in Table 5.

Subgraph transformation For the B^- transformation $F(u,v) \Leftrightarrow E(u,v) \wedge E(v,u)$ holds. This yields the following implications:

$$F(u,v) \Rightarrow E(u,v) \wedge E(v,u) \tag{43}$$
$$\neg F(u,v) \Rightarrow \neg E(u,v) \vee \neg E(v,u) \tag{44}$$
$$\neg E(u,v) \Rightarrow \neg F(u,v) \tag{45}$$

Furthermore, the transformation is symmetric. Therefore, (39) or (40) have to be included in the formula.
From (43) with (21) the expression

$$F(u,v) \Rightarrow \Bigg(E(u,v) \wedge \forall x : \Big(E(u,x) \vee \neg \pi^t_{\mathsf{B}}(u,x,v) \Big) \Bigg) \wedge \Bigg(E(v,u) \wedge \forall y : \Big(E(v,y) \vee \neg \pi^t_{\mathsf{B}}(v,y,u) \Big) \Bigg) \tag{46}$$

is derived. Next (20) and (39) are used in (46). Since the transformation is symmetric and (39) is included in the formula for $F(u,v)$, it is enough to list only the condition for $E(u,v)$ in Table 5.

Furthermore, from (44) with (23), the expression

$$\neg F(u,v) \Rightarrow \Bigg(\neg E(u,v) \wedge \forall x : \Big(\neg E(u,x) \vee \neg \pi^t_{\mathsf{B}}(u,v,x) \Big) \Bigg) \vee \Bigg(\neg E(v,u) \wedge \forall y : \Big(\neg E(v,y) \vee \neg \pi^t_{\mathsf{B}}(v,u,y) \Big) \Bigg) \tag{47}$$

is derived. Using (22) and (45) in (47) yields the formula in Table 5.

Empty Graph transformation For the transformation $\mathsf{B}^{\perp}$ no edges can exist, therefore, $F(u,v) \Rightarrow \perp$ and $\neg F(u,v) \Rightarrow \top$ holds.

Negated Graph transformations The formulae for the negated transformations can be obtained by switching the formulae for $F(u,v)$ and $\neg F(u,v)$ and also switching $F(a,b)$ and $\neg F(a,b)$ for any pair of vertices a and b inside the formula.

Part 2 ($\Leftarrow$)**.** A graph $H = (V,F)$ is assumed, such that the edge inclusion and exclusion implications depicted for B^{γ} in Table 5 are satisfied. A graph $G = (V,E)$ is constructed, such that $G \in \mathsf{B}$ and $G^{\gamma} = H$, which then implies $H \in \mathsf{B}^{\gamma}$.

Construction of $G = (V,E)$ is conducted in two steps, a *preprocessing* and a *postprocessing* step. The preprocessing is the same for all graph class transformations, while for the postprocessing the specific conditions for the graph transformations have to be considered. Starting with a configuration, where all $E(u,v)$ are set to true. First, $E(u,v)$ is kept true resp. set to false, where this is enforced or forbidden due to the graph class axioms of class B. Then the remaining edges $E(u,v)$ are set, which have not been determined in preprocessing accordingly, such that G still satisfies B and at the same time satisfies $G^{\gamma} = H$.

Preprocessing (forbidden edges): In order to satisfy graph class axiom (16), for all $u,v \in V$, which satisfy $\pi^e_{\mathsf{B}}(u,v)$, $E(u,v)$ is set to false; in this case, in order to also satisfy the axiom (17), $E(u,y)$ is set to false for all $y \in V$ for which $\pi^t_{\mathsf{B}}(u,v,y)$ is satisfied.

Preprocessing (enforced edges): In order to satisfy graph class axiom (15), for all $u, v \in V$ which satisfy $\pi^i_{\mathsf{B}}(u, v)$, $E(u, v)$ is set to true; in this case, in order to satisfy also axiom (17), $E(u, x)$ is set to true for all $x \in V$ for which $\pi^t_{\mathsf{B}}(u, x, v)$ is satisfied. Soundness of inclusion, exclusion and transfer predicates assures that no edge $E(u, v)$ and $E(u, y)$ is set to true, which was set to false before.

Postprocessing: Now, the implications for the specific transformations are considered.
Indentity transformation For the identity transformation B $F(u, v) \Rightarrow \neg\pi^e_{\mathsf{B}}(u, v)$ holds; therefore, the edge $E(u, v)$ was not set to false in the preprocesing and can be set to true now.

Furthermore, $\neg F(u, v) \Rightarrow \neg\pi^i_{\mathsf{B}}(u, v) \wedge \forall w : \big(\neg F(u, w) \vee \neg\pi^t_{\mathsf{B}}(u, v, w)\big)$ assures that $E(u, v)$ was not set to true in the preprocessing and is also not enforced by the existence of another edge $E(u, w)$, since $F(u, w) \Leftrightarrow E(u, w)$. Therefore, the edge $E(u, v)$ can be set to false now.

Reverse transformation For the transformation $\mathsf{B}^{\leftrightarrow}$, $F(u, v) \Rightarrow \neg\pi^e_{\mathsf{B}}(v, u)$ holds, therefore, the edge $E(v, u)$ was not set to false in the preprocessing and can be set to true now.

Furthermore, $\neg F(u, v) \Rightarrow \neg\pi^i_{\mathsf{B}}(v, u) \wedge \forall w : \big(\neg F(w, v) \vee \neg\pi^t_{\mathsf{B}}(v, u, w)\big)$ assures that $E(v, u)$ was not set to true in the preprocessing and is also not enforced by the existence of another edge $E(v, w)$, since $F(w, v) \Leftrightarrow E(v, w)$. Therefore, the edge $E(v, u)$ can be set to false now.

Supergraph transformation For the transformation B^+ first the implication for $\neg F(u, v)$ is considered, because $E(u, v)$ and $E(v, u)$ have to be set to false for $\neg F(u, v)$. The implication for $\neg F(u, v) \Rightarrow \neg F(v, u) \wedge \neg\pi^i_{\mathsf{B}}(u, v) \wedge \forall w : \big(\neg\pi^i_{\mathsf{B}}(u, w) \vee \neg\pi^t_{\mathsf{B}}(u, v, w)\big)$ assures that $E(u, v)$ was not set to true in the preprocessing and is also not enforced by the existence of other edges. Therefore, the edge $E(u, v)$ can be set to false now. The symmetry of $F(u, v)$ assures that the same holds for $E(v, u)$. Therefore, also $E(v, u)$ can be set to false now.

Now, the implication for $F(u, v)$ is considered and it is assumed first that $F(u, v) \Rightarrow \neg\pi^e_{\mathsf{B}}(u, v) \wedge \forall x : \Big(\big(F(u, x) \wedge \neg\pi^e_{\mathsf{B}}(u, x)\big) \vee \neg\pi^t_{\mathsf{B}}(u, x, v)\Big)$ holds. This implication assures that $E(u, v)$ was not set to false in the preprocessing. Furthermore, $E(u, v)$ can also not be set to false, by the non-existence of an other edge, because if $\pi^t_{\mathsf{B}}(u, x, v)$ holds, then also $F(u, x)$ holds and therefore $E(u, x)$ can be set to true, too. Therefore, $E(u, v)$ can be set to true now. With the same argument, $E(v, u)$ can be set to true if $F(u, v) \Rightarrow \neg\pi^e_{\mathsf{B}}(v, u) \wedge \forall y : \Big(\big(F(u, y) \wedge \neg\pi^e_{\mathsf{B}}(u, y)\big) \vee \neg\pi^t_{\mathsf{B}}(v, y, u)\Big)$ holds.

Subgraph transformation For the transformation B^-, first the implication for $F(u, v)$ is considered, because $E(u, v)$ and $E(v, u)$ have to be set to true for $F(u, v)$. The implication $F(u, v) \Rightarrow \neg\pi^e_{\mathsf{B}}(u, v) \wedge \forall w : \big(\neg\pi^e_{\mathsf{B}}(u, w) \vee \neg\pi^t_{\mathsf{B}}(u, w, v)\big)$ assures that $E(u, v)$ was not set to false in the preprocessing and is also not removed by the non-existence of other edges. Therefore, $E(u, v)$ can be set to true now. The symmetry of $F(u, v)$ assures that the same holds for $E(v, u)$. Therefore, also $E(v, u)$ can be set to true now.

Now, the implication for $\neg F(u, v)$ is considered and it is assumed first that $\neg F(u, v) \Rightarrow \neg\pi^i_{\mathsf{B}}(u, v) \wedge \forall x : \Big(\big(\neg F(u, x) \wedge \neg\pi^i_{\mathsf{B}}(u, x)\big) \vee \neg\pi^t_{\mathsf{B}}(u, v, x)\Big)$ holds. This implication assures that $E(u, v)$ was not set to true in the preprocessing. Furthermore, $E(u, v)$ can also not be set

to true, by the existence of any other edge, because if $\pi^t_{\mathsf{B}}(u,v,x)$ holds, then also $\neg F(u,x)$ holds and therefore $E(u,x)$ can be set to false, too. Therefore, $E(u,v)$ can be set to true now. With the same argument $E(v,u)$ can be set to false if $\neg F(u,v) \Rightarrow \neg\pi^i_{\mathsf{B}}(v,u) \wedge \forall y : \Big(\big(\neg F(v,y) \wedge \neg\pi^i_{\mathsf{B}}(v,y)\big) \vee \neg\pi^t_{\mathsf{B}}(v,u,y)\Big)$ holds.

Empty graph transformation For the transformation $\mathsf{B}^{\perp}$, no transformation specific restrictions have to be considered, because every graph is transformed to the empty graph. Therefore, after the preprocessing, the remaining edges can be set to true or false arbitrarily.

Negated graph transformations The proofs for the negated transformations can be obtained analogously by replacing systematically $F(a,b)$ with $\neg F(a,b)$ and vice versa for all vertices a, b occurring in the proof. □

By negating the implications in Table 5 and using skolemization for the existentially quantified vertices, one obtains a formula expressing properties of up to $\mu(\gamma)$ vertices.
After deriving the conditions for the existence resp. non-existence of edges in a transformed graph, the following task is considered next: Two graph classes, A and B, are defined by inclusion, exclusion and transfer axioms, and two transformations γ_1 and γ_2. It is considered whether $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{B})$ or $\gamma_1(\mathsf{A}) \not\subseteq \gamma_2(\mathsf{B})$.

In general, $\mathsf{A} \not\subseteq \gamma(\mathsf{B})$, if there exists a graph $G = (V,E) \in \mathsf{A}$, which is not contained in $\gamma(\mathsf{B})$. Such a graph is termed *witness graph*. Moreover, G is termed a *minimum witness graph* if there exists no witness graph $G' = (V',E')$ with $V' \subsetneq V$. In general, a minimum witness graph is not necessarily unique.

The size of a minimum witness graph depends on the transformation γ and the transformed class B. As summarized in the next lemma, for the simple vertex preserving transformations in Γ_1, with Lemma 5 upper bounds on the number of vertices of a *minimum witness graph* are obtained. This value is termed the *maximum size of a minimum witness graph* $\mu(\gamma)$.

Lemma 6. *Let γ_1 and γ_2 be any of the vertex preserving simple graph transformations in Γ_1. Let A and B be graph classes defined by inclusion, exclusion and transfer axiom, using predicates satisfying Independency, Transitivity and Soundness (Assumptions 1–3). Let $\mu(\gamma_2)$ as defined in Table 5 for transformation γ_2. $\gamma_1(\mathsf{A}) \not\subseteq \gamma_2(\mathsf{B})$ if there exists a graph $G = (V,E)$ with $V \subseteq \Omega$ and $|V| \leq \mu(\gamma_2)$, such that $G \in \gamma_1(\mathsf{A})$ and $G \notin \gamma_2(\mathsf{B})$.*

Proof. With $\gamma_1(\mathsf{A}) \not\subseteq \gamma_2(\mathsf{B})$ it follows that there exists a graph $G = (V,E)$ with $V \subseteq \Omega$ such that $G \in \gamma_1(\mathsf{A})$ and $G \notin \gamma_2(\mathsf{B})$. Consider such graph $G = (V,E)$, which is minimal in $|V|$. Assume $|V| > \mu(\gamma_2)$.

Since $G \notin \gamma_2(\mathsf{B})$, Lemma 5 implies that there exist $u_1, \ldots, u_{\mu(\gamma_2)}$, such that at least one of the expressions for $F(u,v)$ or $\neg F(u,v)$ for transformation γ_2 in Table 5 is not satisfied.

Since $|V| > \mu(\gamma_2)$, there exists a $y_0 \in V$, which differs from $u_1, \ldots, u_{\mu(\gamma_2)}$. Consider the subgraph G' induced by $V \setminus \{y_0\}$. All edges between vertices of $u_1, \ldots, u_{\mu(\gamma_2)} \setminus y_0$ exist

in G' iff they exist in G. Thus, at least one of the expressions for $F(u,v)$ or $\neg F(u,v)$ for transformation γ_2 does also not hold for G' and thus G' is not contained in $\gamma_2(\mathsf{B})$.

However, with Lemma 3 the subgraph G' induced by $V \setminus \{y_0\}$ is a graph in $\gamma_1(\mathsf{A})$. Since G was assumed to be minimal in $|V|$, graph G' must be contained in $\gamma_2(\mathsf{B})$ (otherwise G would not be a minimum witness graph), a contradiction. □

It is next regarded if similar results as in Lemma 5 and Lemma 6 can also be obtained for the transformations in Γ_2 and Γ_3

4.3.3 Axiomatization for the Graph Classes Obtained by Transformations in Γ_2

In this section graph classes obtained by a transformation in Γ_2, i.e. the transformations $(\cdot)^{\rightarrow}$, $(\cdot)^{\leftarrow}$, $(\cdot)^{\neg\rightarrow}$ and $(\cdot)^{\neg\leftarrow}$ are analyzed. For the transformations in Γ_2 it is first proven that without further assumptions on the predicates defining a graph class, a minimal witness graph can be arbitrary large. Afterwards additional assumptions are made, in order to derive some necessary conditions for graphs obtained by the transformations in Γ_2. Those conditions are used to prove that certain small patterns of a graph can be contained in a graph obtained by a transformation in Γ_2. However, in general it is not guaranteed that a graph that satisfies the conditions, can also be a graph obtained by a transformation in Γ_2.
First graphs obtained by a transformation in Γ_2 applied on a graph defined by predicates satisfying Assumptions 1-3 are considered. It is investigated, if the maximum size of a minimum witness graph can be arbitrary large, for those graphs.

Lemma 7. *The maximum size of a minimum witness graph for the transformations in Γ_2 is unbounded.*

Proof. Assume the maximum size of a minimum witness graph for the transformation $(\cdot)^{\rightarrow}$ is k. In order to obtain a contradiction it is proven that for a vertex universe Ω there exists a graph $H = (V, F)$ with $V \subseteq \Omega$ and $|V| = k+1$, which is a minimal witness graph. For this the graph class A of all graphs with an empty set of edges and the graph class $\mathsf{B}^{\rightarrow}$ are regarded. Let the vertex set be $V = \{v_1, \ldots, v_{k+1}\}$ and the graph class B be defined such that the predicates $\pi^i_{\mathsf{B}}(v_1, v_2)$ and $\pi^e_{\mathsf{B}}(v_{k+1}, v_k)$ are true. Furthermore, all predicates $\pi^t_{\mathsf{B}}(v_j, v_{j+1}, v_{j-1})$ are true for j and $2 \leq j \leq k$ (see Fig. 10(**a**)). All other predicates are false.

Consider the graph $H = (V, F) \in \mathsf{A}$, where the edge set F is empty. It is proven that no graph $G \in \mathsf{B}$ exists such that $H = G^{\rightarrow}$. In order to obtain a contradiction it is now regarded, which edges have to be set to true resp. false in G. First the edge $E(v_1, v_2)$ is set to true, in order to satisfy the inclusion axiom. Therefore, also the edge $E(v_2, v_1)$ has to be set to true, because otherwise the edge $F(v_1, v_2)$ would exist in the transformed graph. Since the transfer predicate $\pi^t_{\mathsf{B}}(v_2, v_3, v_1)$ is true also the edge $E(v_2, v_3)$ has to be set to true. With the same argument and since all transfer predicates $\pi^t_{\mathsf{B}}(v_j, v_{j+1}, v_{j-1})$ are true for j and $2 \leq j \leq k$ all the edges $E(v_i, v_{i+1})$ and $E(v_{i+1}, v_i)$ have to be set to true for $i = \{1, ..., k\}$.

However, the edge $E(v_{k+1}, v_k)$ has to be set to false in order to satisfy the exclusion predicate, a contradiction. Therefore, no graph $G \in \mathsf{B}$ exists, such that $H = G^{\rightarrow}$. Therefore, H is a witness graph proving that $\mathsf{A} \not\subseteq \mathsf{B}^{\rightarrow}$.

It is next proven that H is a minimum witness graph. For this it is proven that there exists no witness graph $H' = (V', F')$ with $V' \subseteq \Omega$ and $|V| \leq k$. Since $H' \in \mathsf{A}$, the edge set F' is empty for any graph H'. Depending on the vertex set V' the edges of E are set to true resp. false, in order to obtain a graph $G = (V', E)$, such that $H' = G^{\rightarrow}$. If v_1 or v_2 are not in V', then all edges of G can be set to false. Otherwise let $i' = \min\{i | v_i \notin V'\}$. Then $E(v_i, v_{i+1})$ and $E(v_{i+1}, v_i)$ are set to true for all $i < i' - 1$ and all other edges are set to false. Therefore, for every graph $H' = (V', F')$ with $|V'| \leq k$, there exists a graph $G = (V', E)$, such that $H' = G^{\rightarrow}$. Since, the minimum size of a maximum witness graph is $k + 1$ and since k was arbitrary, the maximum size of a minimum witness graph is unbounded.

A similar construction yields that also the maximum size of a minimum witness graph for the transformations $(\cdot)^{\leftarrow}$ (see Fig. 10(**b**)), $(\cdot)^{\neg\rightarrow}$ and $(\cdot)^{\neg\leftarrow}$ is unbounded. □

(**a**) After applying the $(\cdot)^{\rightarrow}$ transformation one of the edges $F(v_i, v_{i+1})$ has to exist

(**b**) After applying the $(\cdot)^{\leftarrow}$ transformation one of the edges $F(v_i, v_{i+1})$ has to exist

Figure 10: Illustration of the counterexamples from Lemma 7.

The situation described in the proof of Lemma 7 can be avoided under additional assumptions.

Assumption 4. (Symmetry) *The inclusion and exclusion predicates are symmetric, i.e.* $\forall u, v \in V : \pi^i_{\mathsf{A}}(u, v) \Leftrightarrow \pi^i_{\mathsf{A}}(v, u)$ *and* $\pi^e_{\mathsf{A}}(u, v) \Leftrightarrow \pi^e_{\mathsf{A}}(v, u)$ *holds.*

Assumption 5. (Simplication) *A graph class* A *has simplified predicates if* $\forall u, v, w \in V :$ $\pi^i_{\mathsf{A}}(u, v) \wedge \pi^t_{\mathsf{A}}(u, w, v) \Rightarrow \pi^i_{\mathsf{A}}(u, w)$ *and* $\pi^e_{\mathsf{A}}(u, v) \wedge \pi^t_{\mathsf{A}}(u, v, w) \Rightarrow \pi^e_{\mathsf{A}}(u, w)$ *holds.*

It is easy to see, as long as $r_1 \leq r_2$ any conjunction of the predicates used in the axioms for $\mathsf{MinDG}(r_1)$, $\mathsf{MaxDG}(r_2)$ and CRG satisfies the satisfies Assumptions 4 and 5. Moreover, every graph class, where all inclusion and all exclusion predicates are false satisfies Assumptions 4 and 5. Under Assumption 4 and 5 it is always possible to choose the edge configuration of a graph $G = (V, E)$, such that the graph G is symmetric. Since for a symmetric graph $G = (V, E)$, the graph $H = G^{\rightarrow}$ is always the empty graph, the construction in Lemma 7 is not possible anymore under Assumption 4 and 5.
Next the connection between edge configurations $F(u, v)$ for a graph being in a class obtained by a transformation in Γ_2 and the inclusion, exclusion and transfer predicates of that class is considered.

Lemma 8. *Let* B *be a graph class described by inclusion, exclusion and transfer predicates* $\pi^i_{\mathsf{B}}(u,v)$, $\pi^e_{\mathsf{B}}(u,v)$, $\pi^t_{\mathsf{B}}(u,v,w)$, *satisfying Assumptions 1-5. Let* γ *be one of the vertex preserving, simple graph transformations in* Γ_2. *For a graph* $H=(V,F)$ *in class* $\gamma(\mathsf{B})$ *it holds that, for every* $u,v \in V$ *the edge relation* $F(u,v)$ *satisfies the implications depicted in Table 6 for that transformation.*

γ	**Edge Inclusion and Exclusion Implications**		
$H \in \mathsf{B}^{\rightarrow}$	$F(u,v)$	$\Rightarrow$	$\neg F(v,u) \wedge \Big(\neg\pi^e_{\mathsf{B}}(u,v) \wedge \forall x : \Big(\neg F(x,u) \vee \neg\pi^t_{\mathsf{B}}(u,x,v)\Big)\Big) \wedge \Big(\neg\pi^i_{\mathsf{B}}(v,u) \wedge \forall y : \Big(\neg F(v,y) \vee \neg\pi^t_{\mathsf{B}}(v,u,y)\Big)\Big)$
	$\neg F(u,v)$	$\Rightarrow$	$\forall x : \Big(\neg F(u,x) \vee \neg\pi^t_{\mathsf{B}}(u,v,x)\Big) \vee \forall y : \Big(\neg F(y,v) \vee \neg\pi^t_{\mathsf{B}}(v,y,u)\Big)$
$H \in \mathsf{B}^{\neg\rightarrow}$	$F(u,v)$	$\Rightarrow$	$\forall x : \Big(F(u,x) \vee \neg\pi^t_{\mathsf{B}}(u,v,x)\Big) \vee \forall y : \Big(F(y,v) \vee \neg\pi^t_{\mathsf{B}}(v,y,u)\Big)$
	$\neg F(u,v)$	$\Rightarrow$	$F(v,u) \wedge \Big(\neg\pi^e_{\mathsf{B}}(u,v) \wedge \forall x : \Big(F(x,u) \vee \neg\pi^t_{\mathsf{B}}(u,x,v)\Big)\Big) \wedge \Big(\neg\pi^i_{\mathsf{B}}(v,u) \wedge \forall y : \Big(F(v,y) \vee \neg\pi^t_{\mathsf{B}}(v,u,y)\Big)\Big)$
$H \in \mathsf{B}^{\leftarrow}$	$F(u,v)$	$\Rightarrow$	$\neg F(v,u) \wedge \Big(\neg\pi^e_{\mathsf{B}}(v,u) \wedge \forall x : \Big(\neg F(v,x) \vee \neg\pi^t_{\mathsf{B}}(v,x,u)\Big)\Big) \wedge \Big(\neg\pi^i_{\mathsf{B}}(u,v) \wedge \forall y : \Big(\neg F(y,u) \vee \neg\pi^t_{\mathsf{B}}(u,v,y)\Big)\Big)$
	$\neg F(u,v)$	$\Rightarrow$	$\forall x : \Big(\neg F(x,v) \vee \neg\pi^t_{\mathsf{B}}(v,u,x)\Big) \vee \forall y : \Big(\neg F(u,y) \vee \neg\pi^t_{\mathsf{B}}(u,y,v)\Big)$
$H \in \mathsf{B}^{\neg\leftarrow}$	$F(u,v)$	$\Rightarrow$	$\forall x : \Big(F(x,v) \vee \neg\pi^t_{\mathsf{B}}(v,u,x)\Big) \vee \forall y : \Big(F(u,y) \vee \neg\pi^t_{\mathsf{B}}(u,y,v)\Big)$
	$\neg F(u,v)$	$\Rightarrow$	$F(v,u) \wedge \Big(\neg\pi^e_{\mathsf{B}}(v,u) \wedge \forall x : \Big(F(v,x) \vee \neg\pi^t_{\mathsf{B}}(v,x,u)\Big)\Big) \wedge \Big(\neg\pi^i_{\mathsf{B}}(u,v) \wedge \forall y : \Big(F(y,u) \vee \neg\pi^t_{\mathsf{B}}(u,v,y)\Big)\Big)$

Table 6: The expressions implied by edge existence and non-existence under the considered vertex preserving simple graph class transformations. Quantification is over all vertices V of the considered graph $H=(V,F)$.

Proof. The proof follows the same structure as the proof of Part1 of Lemma 5. All parts of the proof that are not transformation specific are analogous to the proof of Lemma 5. Therefore, only the transformation specific parts of the proof have to be considered.
Outgoing transformation For the $\mathsf{B}^{\rightarrow}$ transformation $F(u,v) \Leftrightarrow E(u,v) \wedge \neg E(v,u)$ holds. This yields the following implications:

$$F(u,v) \Rightarrow E(u,v) \wedge \neg E(v,u) \tag{48}$$
$$\neg F(u,v) \Rightarrow \neg E(u,v) \vee E(v,u) \tag{49}$$
$$E(v,u) \Rightarrow \neg F(u,v) \tag{50}$$
$$\neg E(u,v) \Rightarrow \neg F(u,v) \tag{51}$$

Furthermore, the transformation is oriented, therefore the following implication has to be included in the formula:

$$F(u,v) \Rightarrow \neg F(v,u) \tag{52}$$

From (48) with (21) and (23)

$$\begin{aligned} F(u,v) \Rightarrow & \left(E(u,v) \wedge \forall x : \left(E(u,x) \vee \neg \pi^t_{\mathsf{B}}(u,x,v) \right) \right) \wedge \\ & \left(\neg E(v,u) \wedge \forall y : \left(\neg E(v,y) \vee \neg \pi^t_{\mathsf{B}}(v,u,y) \right) \right) \end{aligned} \tag{53}$$

is derived. Using (20), (22), (50), (51) and (52) in (53) yields the formula in Table 6.

Furthermore, from (49) with (21) and (23)the expression

$$\neg F(u,v) \Rightarrow \forall x : \left(\neg E(u,x) \vee \neg \pi^t_{\mathsf{B}}(u,v,x) \right) \vee \forall y : \left(E(v,y) \vee \neg \pi^t_{\mathsf{B}}(v,y,u) \right) \tag{54}$$

is derived. Using (50) and (51) in (54) yields the expression in the table.

Incoming transformation For the $\mathsf{B}^{\leftarrow}$ transformation $F(u,v) \Leftrightarrow \neg E(u,v) \wedge E(v,u)$ holds. This yields the following implications:

$$F(u,v) \Rightarrow \neg E(u,v) \wedge E(v,u) \tag{55}$$

$$\neg F(u,v) \Rightarrow E(u,v) \vee \neg E(v,u) \tag{56}$$

$$E(u,v) \Rightarrow \neg F(u,v) \tag{57}$$

$$\neg E(v,u) \Rightarrow \neg F(u,v) \tag{58}$$

Furthermore, the transformation is oriented, therefore (52) has to be included in the formula. From (55) with (21) and (23)

$$\begin{aligned} F(u,v) \Rightarrow & \left(\neg E(u,v) \wedge \forall x : \left(\neg E(u,x) \vee \neg \pi^t_{\mathsf{B}}(u,v,x) \right) \right) \wedge \\ & \left(E(v,u) \wedge \forall y : \left(E(v,y) \vee \neg \pi^t_{\mathsf{B}}(v,y,u) \right) \right) \end{aligned} \tag{59}$$

is derived.

Therefore, using (20), (22), (52), (57) and (58) in (59) yields the formula in Table 6.

Furthermore, from (56) with (21) and (23) the expression

$$\neg F(u,v) \Rightarrow \forall x : \left(E(u,x) \vee \neg \pi^t_{\mathsf{B}}(u,x,v) \right) \vee \forall y : \left(\neg E(v,y) \vee \neg \pi^t_{\mathsf{B}}(v,u,y) \right) \tag{60}$$

is derived.
Therefore, using (57) and (58) in (60) yields the formula in the Table 6.

Negated graph transformations The formulae for the negated transformations can be obtained by switching the formulae for $F(u,v)$ and $\neg F(u,v)$ and also switching $F(a,b)$ and $\neg F(a,b)$ for any pair of vertices a and b inside the formula.

□

By negating the implications in Table 6 and using skolemization for the existentially quantified vertices, one obtains a formula with up to 4 vertices.
In contrast to the conditions derived in Lemma 5 for the graph classes obtained by transformations in Γ_1, the conditions for the graph classes obtained by transformations in Γ_2 in Lemma 8 are not sufficient. Therefore, it is possible that a graph $H = (V, F)$ satisfies the conditions from Lemma 8 for each vertex pair, but there exists no graph G, such that $H = \gamma(G)$. Moreover, the maximal size of a minimum witness graph for a graph class obtained by a transformation in Γ_2 is also unbounded, if the predicates are symmetric and simplified. This is proven by considering a similar example as in the proof of Lemma 7.

Corollary 9. *The maximum size of a minimum witness graph for the transformations in* Γ_2 *is also under Assumptions 4 and 5 unbounded.*

Proof. The example constructed in the proof of Lemma 7 is considered, with all inclusion and exclusion predicates being false. In this case obviously also Assumption 4 and 5 are satisfied. It is investigated whether for the graph $H = (V, F)$ with $F = \{v_2v_1, v_{k+1}v_k\}$, a graph $G = (V, E)$ exists, such that $H = G^{\rightarrow}$. With the same argumentation as in the proof of Lemma 7 it is proven that there exists no graph G, such that $H = G^{\rightarrow}$. Moreover, from the proof of Lemma 7 it follows that H is a minimal witness graph. Similar constructions yield that also for the remaining transformation in Γ_2 the maximum size of a minimum witness graph is unbounded. □

When instead of a graph, only certain graph patterns P are regarded, then the conditions from Lemma 8 are also sufficient to guarantee that there exists a graph H which contains the graph pattern P. In such a graph pattern $P(V, S, T)$ all edges in S must exist, while all edges in T must not exist. The existence of all remaining edges is not specified. Soundness ensures the the edge configurations of those edges not occurring in the graph pattern can be chosen such that all graph axioms are satisfied for the according graph class. In the following, a graph pattern $P(V, S, T)$ corresponds to a triangle if for $u, v, w \in V$: $S \cup T = \{(u, v), (v, w), (w, u)\}$.

Lemma 10. *Let* B *be a graph class defined by axioms using predicates satisfying Assumptions 1-5 and let* γ *be a graph transformation in* Γ_2*. If for a graph pattern* $P = (V, S, T)$ *of size up to 3, which does not correspond to a triangle, for every vertex pair* $(u, v) \in S$ *the implication for* $F(u, v)$ *in Table 6 is satisfied and for every vertex pair* $(u, v) \in T$ *the implication for* $\neg F(u, v)$ *in Table 6 is satisfied, then there exists a graph* $H = (V, F)$ *in* $\gamma(B)$*, such that the graph pattern* P *is contained in the graph* H*.*

Proof. The preprocessing of edges follows in the same way as in the proof of Lemma 5. Therefore, it remains to consider only the postprocessing of the edges.

Outgoing transformation For the transformation $\mathsf{B}^{\rightarrow}$, first the vertex pairs in S are considered, i.e. vertex pairs for which the implication for $F(u, v)$ is satisfied. For this implication, first the part of the implication that assures that $E(u, v)$ can be set to true is considered and afterwards the part that assures that $E(v, u)$ can be set to false.

The implication $F(u, v) \Rightarrow \neg\pi^e_{\mathsf{B}}(u, v) \wedge \forall x : \Big(\neg F(x, u) \vee \neg\pi^t_{\mathsf{B}}(u, x, v)\Big)$ assures that $E(u, v)$ was not set to false in the preprocessing. Furthermore, $E(u, v)$ can also not be set to false, by the existence of an other edge, because if $\pi^t_{\mathsf{B}}(u, x, v)$ holds, then also $\neg F(x, u)$ holds and therefore $E(u, x)$ can be set to true as well. Therefore, $E(u, v)$ can be set to true now.

The implication $F(u, v) \Rightarrow \neg\pi^i_{\mathsf{B}}(v, u) \wedge \forall y : \Big(\neg F(v, y) \vee \neg\pi^t_{\mathsf{B}}(v, u, y)\Big)$ assures that $E(v, u)$ was not set to true in the preprocessing. Furthermore, $E(v, u)$ can as well not be set to true, by the existence of another edge, because if $\pi^t_{\mathsf{B}}(v, u, y)$ holds, then also $\neg F(v, y)$ holds and therefore $E(v, y)$ can be set to false as well. Therefore, $E(v, u)$ can be set to false now. If necessary, further edge configurations in E are set now, to fulfill axioms (17) and (18).

Afterwards, or if the set S of a graph pattern was empty, only the edge configurations of E for vertex pairs in T remains to be chosen. The vertex pairs in T satisfy the implication for $\neg F(u, v)$. Since the vertex pairs are not forming a triangle, there exists at most one vertex pair in T, where both vertices of this vertex pair, occur also in another vertex pair of the graph pattern. If such a vertex pair exists, the edge configuration of E for this vertex pair is chosen next. The implication for $\neg F(u, v)$ assures that $E(u, v)$ can be set to false or $E(v, u)$ can be set to true. If necessary, further edge configurations in E are set now, to fulfill axioms (17) and (18).

Afterwards, or if the graph pattern did not contain a vertex pair in T, where both vertices of the vertex pair also occur in another vertex pair of the graph pattern, for the remaining vertex pairs of the graph pattern, the edge configurations for $E(u, v)$, $E(v, u)$ or for both, $E(u, v)$ and $E(v, u)$, was not chosen so far. Therefore the edge configuration for those edges can be chosen now, such that $E(u, v)$ and $E(v, u)$ are both true or both false.

The edge configuration of E for all vertex pairs, that were not contained in the graph pattern can now be chosen arbitrarily under the condition that (15)- (18) are satisfied. This yields a graph $G = (V, E)$ such that the graph pattern $P(V, S, T)$ is contained in the graph $G^{\rightarrow} = H = (V, F)$.

Incoming transformation For the transformation $\mathsf{B}^{\leftarrow}$, first the vertex pairs in S are considered. The implication for $F(u,v)$ is satisfied for the vertex pairs in S. For this implication, first the part of the implication that assures that $E(v,u)$ can be set to true is considered and afterwards the second part of the implication that assures that $E(u,v)$ can be set to false.

The implication $F(u,v) \Rightarrow \neg\pi^e_{\mathsf{B}}(v,u) \wedge \forall y : \Big(\neg F(v,y) \vee \neg\pi^t_{\mathsf{B}}(v,y,u)\Big)$ assures that $E(v,u)$ was not set to false in the preprocessing. Furthermore, $E(v,u)$ can also not be set to false, by the existence of an other edge, because if $\pi^t_{\mathsf{B}}(v,y,u)$ holds, then also $\neg F(v,y)$ holds and therefore $E(v,y)$ can be set to true as well. Therefore, $E(v,u)$ can be set to true now.

The implication $F(u,v) \Rightarrow \neg\pi^i_{\mathsf{B}}(u,v) \wedge \forall x : \Big(\neg F(x,u) \vee \neg\pi^t_{\mathsf{B}}(u,v,x)\Big)$ assures that $E(u,v)$ was not set to true in the preprocessing. Furthermore, $E(u,v)$ can also not be set to false, by the existence of an other edge, because if $\pi^t_{\mathsf{B}}(u,v,x)$ holds, then also $\neg F(x,u)$ holds and therefore $E(u,x)$ can be set to false as well. Therefore, $E(u,v)$ can be set to false now. If necessary the edge configuration of further edges in E is chosen now, in order to satisfy axioms (17) and (18).

Afterwards, or if the set S of a graph pattern was empty, only the edge configuration of E for vertex pairs in T remains to be chosen. The vertex pairs in T satisfy the implication for $\neg F(u,v)$. Since the vertex pairs are not forming a triangle, there exists at most one vertex pair in T, where both vertices of this vertex pair, occur also in another vertex pair of the graph pattern. If such a vertex pair exists, the edge configuration of E for this vertex pair is chosen next. The implication for $\neg F(u,v)$ assures that $E(u,v)$ can be set to false or $E(v,u)$ can be set to true. If necessary the edge configuration of further edges in E is chosen now, in order to satisfy axioms (17) and (18).

Afterwards, or if the graph pattern did not contain a vertex pair in T, where both vertices of the vertex pair also occur in another vertex pair of the graph pattern, for the remaining vertex pairs of the graph pattern, the edge configuration for $E(u,v)$, $E(v,u)$ or for both, $E(u,v)$ and $E(v,u)$, was not chosen so far. Therefore the edge configuration for $E(u,v)$, $E(v,u)$ or for both, $E(u,v)$ and $E(v,u)$, can be chosen now, such that $E(u,v)$ and $E(v,u)$ are both true or both false.

The edge configuration of E for all vertex pairs, that were not contained in the graph pattern can now be chosen arbitrarily under the condition that (15)- (18) are satisfied. This yields a graph $G = (V,E)$ such that the graph pattern $P(V,S,T)$ is contained in the graph $G^{\leftarrow} = H = (V,F)$.

Negated graph transformations The proofs for the negated transformations can be obtained analogously by replacing systematically $F(a,b)$ with $\neg F(a,b)$ and vice versa for all vertices a, b occurring in the proof. □

4.3.4 Axiomatization for the Graph Classes Obtained by Transformations in Γ_3

In this section graph classes obtained by a transformation in Γ_3, i.e. the transformations $(\cdot)^{\oplus}$ and $(\cdot)^{\neg\oplus}$ are analyzed. Also for the transformations in Γ_3 it is first proven that a minimal witness graph can be arbitrary large. Afterwards some necessary conditions for graphs obtained by the transformations in Γ_3 are derived. Those conditions are used to prove that certain small patterns of a graph can be contained in a graph obtained by a transformation in Γ_3.
First graphs obtained by a transformation in Γ_3 applied on a graph defined by predicates satisfying Assumptions 1-5 are considered. It is investigated, if the maximum size of a minimum witness graph can be arbitrary large, for those graphs.

Lemma 11. *The maximum size of the minimum witness graph for the transformations in* Γ_3 *is unbounded*

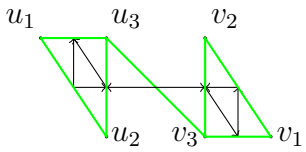
Proof. Assume the maximum size of a minimum witness graph for the transformation $(\cdot)^{\oplus}$ is k. In order to obtain a contradiction it is proven that for a vertex universe Ω there exists a graph $H = (V, F)$ with $V \subseteq \Omega$ and $|V| = m = \lfloor (k+1)/2 \rfloor$, which is a minimal witness graph. For this the graph class A of the complete graphs and the graph class $\mathsf{B}^{\oplus}$ are regarded. Let the vertex set be $V = \{u_1, \ldots, u_m, v_m, \ldots, v_1\}$, such that $u_{m+1} = v_m$ and $v_{m+1} = u_m$. The graph class B is defined such that for all $i \geq 2$ $\pi^t_{\mathsf{B}}(u_i, u_{i-1}, u_{i+1})$ and $\pi^t_{\mathsf{B}}(v_i, v_{i-1}, v_{i+1})$ are true if i is even and $\pi^t_{\mathsf{B}}(u_i, u_{i+1}, u_{i-1})$ and $\pi^t_{\mathsf{B}}(v_i, v_{i+1}, v_{i-1})$ are true if i is odd. Furthermore, $\pi^t_{\mathsf{B}}(v_1, v_2, v_3)$, $\pi^t_{\mathsf{B}}(v_3, v_2, v_1)$, $\pi^t_{\mathsf{B}}(u_1, u_2, u_3)$ and $\pi^t_{\mathsf{B}}(u_3, u_2, u_1)$ are true. All other transfer predicates, as well as all inclusion and exclusion predicates are false. In Fig. 11 examples of graphs for $m = 3$ (Fig. 11(**a**)) and $m = 4$ (Fig. 11(**b**)) are illustrated.

Starting with the edge v_2v_3 it is first assumed that $E(v_2, v_3)$ is true. In order to include v_2v_3 in F, $E(v_3, v_2)$ has to be set to false. In order to satisfy the transfer axiom, $E(v_2, v_1)$ has to be set to true and $E(v_3, v_1)$ has to be set to false. Consequently, $E(v_1, v_2)$ has to be set to false, while $E(v_1, v_3)$ has to be set to true, in order to include the edges v_1v_2 and v_1v_3 in F. However, since $\pi^t_{\mathsf{B}}(v_1, v_2, v_3)$ is true, it is not possible to set $E(v_1, v_2)$ to false, while $E(v_1, v_3)$ is set to true. Therefore, $E(v_2, v_3)$ has to be set to false. With the same argument also $E(u_2, u_3)$ has to be set to false.

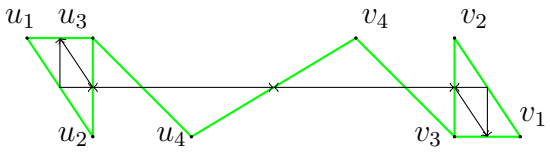
Then $E(u_3, u_2)$ and $E(v_3, v_2)$ have to be set to true in order to include u_2u_3 resp. v_2v_3 in F. In order to satisfy the transfer axiom also $E(u_3, u_4)$ and $E(v_3, v_4)$ have to be set to true. Next $E(u_4, u_3)$ and $E(v_4, v_3)$ have to be set to false, in order to include u_3u_4 and v_3v_4 in F. Consequently $E(u_4, u_5)$ and $E(v_4, v_5)$ have to be set to false in order to satisfy the transfer axiom.

Following this process, the remaining edges $E(u_i, u_{i+1})$ resp. $E(v_i, v_{i+1})$ are set to true for i odd, while they are set to false for i even. Therefore, either $E(u_m, u_{m+1})$ and $E(v_m, v_{m+1})$ are both set to true or both set to false. Consequently, the edge u_mv_m cannot exist in F.

Since all other predicates are false, for any subset of the vertices of V, no contradiction is obtained. Therefore, H is a minimal witness graph of size $2m = 2 \cdot \lfloor (k+1)/2 \rfloor \geq k$. Furthermore, k was chosen arbitrary but fixed and therefore the maximum size of a minimum witness graph for the XOR transformation is unbounded. The same example, with A as the class of empty graphs can be used to prove that also the maximum size of a minimum witness graph for the negated XOR transformation is unbounded. □



(**a**) Witness graph with 6 vertices, in which after applying the $(\cdot)^{\oplus}$ transformation not all of the green edges can exist



(**b**) Witness graph with 8 vertices, in which after applying the $(\cdot)^{\oplus}$ transformation not all of the green edges can exist

Figure 11: Illustration of the counterexamples from Lemma 11

Lemma 12. *Let* B *be a graph class described by inclusion, exclusion and transfer predicates* $\pi^i_{\mathsf{B}}(u,v)$, $\pi^e_{\mathsf{B}}(u,v)$, $\pi^t_{\mathsf{B}}(u,v,w)$, *satisfying Assumption 1-5. Let* γ *be one of the vertex preserving, simple graph transformations in* Γ_3. *For a graph* $H = (V,F)$ *in class* $\gamma(\mathsf{B})$, *for every* $u, v \in V$ *the edge relation* $F(u,v)$ *satisfies the implications depicted in Table 7 for that transformation.*

γ	**Edge Inclusion and Exclusion Implications**		
$H \in \mathsf{B}^{\oplus}$	$F(u,v)$	$\Rightarrow$	$\big(\neg\pi^i_{\mathsf{B}}(u,v) \vee \neg\pi^e_{\mathsf{B}}(u,v)\big)$
	$\neg F(u,v)$	$\Rightarrow$	$\neg F(v,u)$
$H \in \mathsf{B}^{\neg\oplus}$	$F(u,v)$	$\Rightarrow$	$F(v,u)$
	$\neg F(u,v)$	$\Rightarrow$	$\big(\neg\pi^i_{\mathsf{B}}(u,v) \vee \neg\pi^e_{\mathsf{B}}(u,v)\big)$

Table 7: The expressions implied by edge existence and non-existence under the considered vertex preserving simple graph class transformations in Γ_3. Quantification is over all vertices V of the considered graph $H = (V,F)$.

Proof. The proof follows the same structure as the proof of Lemma 5. All parts of the proof that are not transformation specific are analogous to the proof of Lemma 5. Therefore, only the transformation specific parts of the proof have to be considered.
XOR Transformation For the $\mathsf{B}^{\oplus}$ transformation $F(u,v) \Leftrightarrow E(u,v) \wedge \neg E(v,u) \vee \neg E(u,v) \wedge E(v,u)$ holds. This yields the following implication:

$$F(u,v) \Rightarrow \big(E(u,v) \vee E(v,u)\big) \wedge \big(\neg E(u,v) \vee \neg E(v,u)\big) \tag{61}$$

Therefore, using (20) and(22), in (61) yields

$$F(u,v) \Rightarrow \left(\neg\pi^e_{\mathsf{B}}(u,v) \vee \neg\pi^e_{\mathsf{B}}(v,u)\right) \wedge \left(\neg\pi^i_{\mathsf{B}}(u,v) \vee \neg\pi^i_{\mathsf{B}}(v,u)\right) \tag{62}$$

Since the predicates are symmetric, (62) can be simplified to the expression in the table.
For $\neg F(u,v)$ only the symmetry of F is relevant. Other relevant implications already follow from the assumption that the predicates are symmetric and simplified.

Negated XOR transformation The formulae for the negated XOR transformation can be obtained by switching the formulae for $F(u,v)$ and $\neg F(u,v)$ and also switching $F(a,b)$ and $\neg F(a,b)$ for any pair of vertices a and b inside the formula. □

By negating the implications in Table 7 and using skolemization for the existentially quantified vertices, one obtains a formula with up to 2 vertices.
As already proven in Lemma 11 the maximum size of a minimum witness graph for transformations in Γ_3 is unbounded. However, a similar result as stated in Lemma 10 for transformation in Γ_2 does also hold for transformations in Γ_3.

Lemma 13. *Let* B *be a graph class defined by predicates satisfying Assumptions 1-5 and let* γ *be a graph transformation in* Γ_3*. If for a graph pattern* $P = (V, S, T)$ *of size up to 3, which does not correspond to a triangle, for every vertex pair* $(u,v) \in S$ *the implication for* $F(u,v)$ *in Table 7 is satisfied and for every vertex pair* $(u,v) \in T$ *the implication for* $\neg F(u,v)$ *in Table 7 is satisfied, then there exists a graph* $H = (V, F)$ *in* $\gamma(\mathsf{B})$*, such that the graph pattern* P *is contained in the graph* H*.*

Proof. The preprocessing of edges follows in the same way as in the proof of Lemma 5. Therefore, it remains to consider only the postprocessing of the edges.
XOR transformation Since the graph pattern does not correspond to a triangle, there exists at most one vertex pair, where both vertices of this vertex pair, occur also in another vertex pair of the graph pattern. If such a vertex pair exists, and the edge configuration of E for this vertex pair was not already chosen in the preprocessing, then the edge configuration of E for this vertex pair is chosen next. For such a vertex pair, $E(u,v)$ was neither set to true nor false in the preprocessing, and also $E(v,u)$ was neither set to true nor false in the preprocessing. Therefore, $E(u,v)$ can be set to true now. For a vertex pair in S, $E(v,u)$ is set to false now, while for a vertex pair in T, $E(v,u)$ is set to true now.

After choosing the edge configuration of E for the vertex pair, where both vertices occur also in another vertex pair of the graph pattern, or if no such vertex pair exists, only vertex pairs remain, where the edge configuration for $E(u,v)$, $E(v,u)$ or for both, $E(u,v)$ and $E(v,u)$, was not chosen so far. Therefore the edge configuration for $E(u,v)$, $E(v,u)$ or for both, $E(u,v)$ and $E(v,u)$, can be chosen now. For vertex pairs in S the edge configuration can be chosen such that $E(u,v)$ is true, while $E(v,u)$ is false or vice versa. For vertex pairs in T the edge configuration can be chosen such that $E(u,v)$ and $E(v,u)$ are both true or

both false. The edge configuration of E for all vertex pairs, that were not contained in the graph pattern can now be chosen arbitrarily under the condition that (15)- (18) are satisfied. This yields a graph $G = (V, E)$ such that the graph pattern $P(V, S, T)$ is contained in the graph $G^{\oplus} = H = (V, F)$.

Negated XOR transformation The proofs for the negated XOR transformation can be obtained analogously by replacing systematically $F(a, b)$ with $\neg F(a, b)$ and vice versa for all vertices a, b occurring in the proof.

□

4.3.5 Expressions for Proving Subclass Relations

In order to make the testing for proving containedness relations more efficient, only graph patterns instead of concrete graphs are analyzed. For this the following definition is introduced.

Definition 25 (Witness pattern). *For testing the containedness relation $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{B})$, for a given vertex set V a graph pattern $P(V, S, T)$ is called a witness pattern, if P is not contained in a graph in $\gamma_2(\mathsf{B})$, but P is contained in a graph in $\gamma_1(\mathsf{A})$.*

From definition 25 it follows that every graph $H \in \gamma_1(\mathsf{A})$, which contains a witness pattern for testing the containedness relation $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{B})$, is a witness graph for this containedness relation.

The concept of witness patterns is now used to derive formulae for all simple transformations, such that if these formulae are satisfied, then the considered graph pattern cannot be contained in a graph obtained by applying the investigated transformation.

A combination of these formulae in negated and the non-negated form yields an expression for checking if one graph class is contained in another one or not. If a graph class is not contained in another one, then the expression yields conditions for a witness pattern. Such a witness pattern can afterwards be extended to a witness graph by choosing an edge configuration for the remaining edges, such that none of the expressions from the table for the according graph transformation is satisfied for the chosen edge configuration.

In the Tables 8– 12, only the expressions for the non-negated transformations are listed. The expressions for negated transformations can be obtained by swapping the sets S and T of the graph pattern. With the color encoding defined in Figure 9 some of the expressions are illustrated in Figure 12 and Figure 13.

Theorem 14. *Let γ_1 and γ_2 be two of the vertex preserving simple graph transformations. Depending on the transformations γ_1 and γ_2 the following graph classes are considered:*

(a) For two transformations γ_1 and γ_2 in Γ_1 let A and B be graph classes defined by axioms using inclusion, exclusion and transfer predicates satisfying Independence, Transitivity and Soundness (Assumptions 1–3).

(b) For two transformations γ_1 and γ_2 in $\Gamma_1 \cup \Gamma_2 \cup \Gamma_3$ let A and B be graph classes defined by axioms using inclusion, exclusion and transfer predicates satisfying Independence, Transitivity, Soundness, Symmetry and Simplification (Assumptions 1–5).

Then the following holds:

(1) For (a) and (b) let $\rho^k_{\gamma_1(\mathsf{A})}(S^k, T^k, V)$ resp. $\rho^k_{\gamma_2(\mathsf{B})}(S^k, T^k, V)$ be the expressions listed in Tables 8 – 12 for γ_1 and γ_2. It holds: $\gamma_1(\mathsf{A}) \not\subseteq \gamma_2(\mathsf{B})$ if there exists a graph $H = (V, F)$ such that

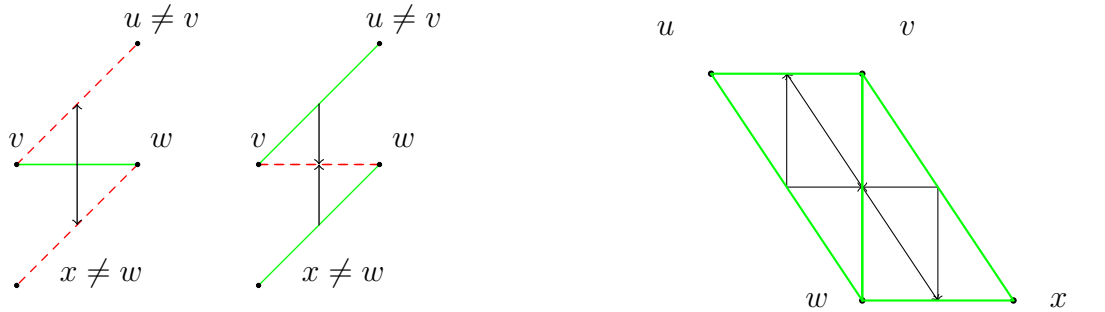
$$\bigvee_{k=1}^{13} \neg\rho^k_{\gamma_1(\mathsf{A})}(S^k, T^k, V) \wedge \rho^k_{\gamma_2(\mathsf{B})}(S^k, T^k, V) \tag{63}$$

is satisfied, where $S^k \subseteq F$, $T^k \subseteq \overline{F}$ and V is the vertex set of H, as defined in the Tables 8 – 12.

(2) For (a) let $\mu(\gamma_2)$ be the maximum size of a minimum witness graph as defined for γ_2 in Table 5. It holds that $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{B})$, if there exists no graph $H = (V, F)$ with $V \leq \mu(\gamma_2)$ satisfying (63).

(**a**) $\rho^1_{\mathsf{B}^+}(\{uv\}, \{\emptyset\})$ (**b**) $\rho^2_{\mathsf{B}^-}(\{\emptyset\}, \{uv\})$

Figure 12: Counterexamples for including and excluding a single edge uv.



(**a**) $\rho^{11}_{\mathsf{B}^+}(\{vw\}, \{uv, wx\})$ and $\rho^{10}_{\mathsf{B}^-}(\{uv, wx\}, \{vw\})$.

(**b**) $\rho^{12}_{\mathsf{B}^\oplus}(\{vw, wu, uv, wv, vx, xw\}, \{\emptyset\})$

Figure 13: Counterexamples for including and excluding more than one edge.

Proof. To obtain a graph H such that $H \in \gamma_1(\mathsf{A})$, but $H \notin \gamma_2(\mathsf{B})$, for the expressions listed in Tables 8 – 12, $\rho^k_{\gamma_1(\mathsf{A})}(S^k, T^k, V)$ has to be unsatisfiable, but $\rho^k_{\gamma_2(\mathsf{B})}(S^k, T^k, V)$ has to be satisfiable. $G = (V, E)$ corresponds to the graph before considering the graph transformations

and $\rho^k_{\mathsf{X}}(S^k, T^k, V)$ means that there exists no graph containing the graph pattern $P(V, S^k, T^k)$. Consequently, $\neg\rho^k_{\mathsf{X}}(S^k, T^k, V)$ means that there exists a graph containing the graph pattern $P(V, S^k, T^k)$

(1) The expressions can be derived by the conjunction of the expressions for graph class transformation X in Tables 5-7. For the vertex pairs in S_k, the expression for $F(u, v)$ has to be used, while for the vertex pairs in T_k the expression for $\neg F(u, v)$ has to be used. The expressions derived in this way refer to the case that a graph pattern $P(V, S^k, T^k)$ is contained in a graph for the according transformation. Therefore, the expression $\neg\rho^k_{\mathsf{X}}(S^k, T^k, V)$ refers to the case that a graph pattern $P(V, S^k, T^k)$ is contained in a graph for the according transformation. Negating $\neg\rho^k_{\mathsf{X}}(S^k, T^k, V)$ yields the expression for $\rho^k_{\mathsf{X}}(S^k, T^k, V)$, given in Tables 8– 12 . The expressions for the negated transformations can be obtained by swapping the sets S_k and T_k. For the transformations in Γ_3, an additional expression, which corresponds to the example constructed in the proof of Lemma 11 for $m = 2$ is added. Since for all other transformations, $\rho^{12}_{\mathsf{X}}(S^{12}, T^{12}, V)$ and $\rho^{13}_{\mathsf{X}}(S^{13}, T^{13}, V)$ can be directly derived by the other expressions for this transformation, they are omitted in the corresponding tables.

(2) Theorem 6 states that $\gamma_1(\mathsf{A}) \not\subseteq \gamma_2(\mathsf{B})$ is satisfied if there exists a counter example with size at most $\mu(\gamma_2)$. Moreover, from the implications derived it follows, that it is sufficient to consider only witnesses patterns $P(V, S, T)$, where S resp. T contain a single vertex pair and the other set is empty, and witness patterns of size up to $\max(2, \mu(\gamma))$, where every vertex pair in S has a vertex in common with a vertex pair in T and vice versa. All those witness patterns are considered in the $\neg\rho^k_{\mathsf{X}}(S^k, T^k, V)$ expressions for the transformations in Γ_1. Therefore, testing the satisfiability of equation (63) is sufficient for testing the containedness relation $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{B})$. □

Under Assumptions 1-5 only the expressions for $\rho^k_{\mathsf{X}}(S^k, T^k, V)$ with $V = V(S^k) \cup V(T^k)$ have to be considered. Furthermore, some of the other expressions also simplify, since they contain conditions, which can only occur if the predicates are not symmetric resp. not simplified.
The expression of Theorem 14 simplifies significantly when testing $\mathsf{A}^- \not\subseteq \mathsf{A}^+$ or $\mathsf{A}^+ \not\subseteq \mathsf{A}^-$ for predicates satisfying Assumptions 1-5.

Corollary 15. *Let* A *be graph class defined by inclusion, exclusion and transfer axioms, using predicates satisfying Assumptions 1-5. Then:* $\mathsf{A}^- \not\subseteq \mathsf{A}^+$ *iff there exist* $u, v, w, x \in V$ *satisfying*

$$\begin{aligned}\neg\pi^e_{\mathsf{A}}(v, w) \wedge \neg\pi^i_{\mathsf{A}}(u, v) \wedge \neg\pi^i_{\mathsf{A}}(w, x) \wedge \\ \pi^t_{\mathsf{A}}(v, u, w) \wedge \pi^t_{\mathsf{A}}(w, x, v)\end{aligned} \tag{64}$$

and $\mathsf{A}^+ \not\subseteq \mathsf{A}^-$ *iff there exist* $u, v, w, x \in V$ *satisfying*

$$\begin{aligned}\neg\pi^i_{\mathsf{A}}(v, w) \wedge \neg\pi^e_{\mathsf{A}}(u, v) \wedge \neg\pi^e_{\mathsf{A}}(w, x) \wedge \\ \pi^t_{\mathsf{A}}(v, w, u) \wedge \pi^t_{\mathsf{A}}(w, v, x)\end{aligned} \tag{65}$$

γ	**Edge Inclusion and Exclusion Implications**		
$(\cdot)^{\perp}$	$\rho^{1}_{\mathsf{B}^{\perp}}(\{uv\},\{\emptyset\},V)$	$=$	$u \neq v$
	$\rho^{2}_{\mathsf{B}^{\perp}}(\{\emptyset\},\{uv\},V)$	$=$	false
$(\cdot)^{\mathrm{id}}$	$\rho^{1a}_{\mathsf{B}}(\{uv\},\{\emptyset\},\{u,v\})$	$=$	$\pi^{e}_{\mathsf{B}}(u,v)$
	$\rho^{1b}_{\mathsf{B}}(\{uv\},\{\emptyset\},\{u,v,w\})$	$=$	$\rho^{1a}_{\mathsf{B}}(\{uv\},\{\emptyset\},\{u,v\}) \vee \Big(\pi^{e}_{\mathsf{B}}(u,w) \wedge \pi^{t}_{\mathsf{B}}(u,w,v)\Big)$
	$\rho^{2a}_{\mathsf{B}}(\{\emptyset\},\{uv\},\{u,v\})$	$=$	$\pi^{i}_{\mathsf{B}}(u,v)$
	$\rho^{2b}_{\mathsf{B}}(\{\emptyset\},\{uv\},\{u,v,w\})$	$=$	$\rho^{2a}_{\mathsf{B}}(\{\emptyset\},\{uv\},\{u,v\}) \vee \Big(\pi^{i}_{\mathsf{B}}(u,w) \wedge \pi^{t}_{\mathsf{B}}(u,v,w)\Big)$
	$\rho^{3}_{\mathsf{B}}(\{uv,vu\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}}(\{vu\},\{\emptyset\},V)$
	$\rho^{4}_{\mathsf{B}}(\{\emptyset\},\{uv,vu\},V)$	$=$	$\rho^{2}_{\mathsf{B}}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}}(\{\emptyset\},\{vu\},V)$
	$\rho^{5}_{\mathsf{B}}(\{uv\},\{vu\},V)$	$=$	$\rho^{1}_{\mathsf{B}}(\{uv\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}}(\{\emptyset\},\{vu\},V)$
	$\rho^{6}_{\mathsf{B}}(\{uv\},\{uw\},V)$	$=$	$\rho^{1}_{\mathsf{B}}(\{uv\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}}(\{\emptyset\},\{uw\},V) \vee$ $\Big(v = w \wedge u \neq v\Big) \vee \pi^{t}_{\mathsf{B}}(u,w,v)$
	$\rho^{7}_{\mathsf{B}}(\{vu\},\{wu\},V)$	$=$	$\rho^{1}_{\mathsf{B}}(\{vu\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}}(\{\emptyset\},\{wu\},V) \vee \Big(v = w \wedge u \neq v\Big)$
	$\rho^{8}_{\mathsf{B}}(\{uv,vw\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}}(\{vw\},\{\emptyset\},V)$
	$\rho^{9}_{\mathsf{B}}(\{\emptyset\},\{uv,vw\},V)$	$=$	$\rho^{2}_{\mathsf{B}}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}}(\{\emptyset\},\{vw\},V)$
	$\rho^{10}_{\mathsf{B}}(\{uv,wx\},\{wv\},V)$	$=$	$\rho^{7}_{\mathsf{B}}(\{uv\},\{wv\},V) \vee \rho^{6}_{\mathsf{B}}(\{wx\},\{wv\},V)$
	$\rho^{11}_{\mathsf{B}}(\{wv\},\{uv,wx\},V)$	$=$	$\rho^{7}_{\mathsf{B}}(\{wv\},\{uv\},V) \vee \rho^{6}_{\mathsf{B}}(\{wv\},\{wx\},V)$
$(\cdot)^{\leftrightarrow}$	$\rho^{1a}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{\emptyset\},\{u,v\})$	$=$	$\pi^{e}_{\mathsf{B}}(v,u)$
	$\rho^{1b}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{\emptyset\},\{u,v,w\})$	$=$	$\rho^{1a}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{\emptyset\},\{u,v\}) \vee \Big(\pi^{e}_{\mathsf{B}}(v,w) \wedge \pi^{t}_{\mathsf{B}}(v,w,u)\Big)$
	$\rho^{2a}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{uv\},\{u,v\})$	$=$	$\pi^{i}_{\mathsf{B}}(v,u)$
	$\rho^{2b}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{uv\},\{u,v,w\})$	$=$	$\rho^{2a}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{uv\},\{u,v\}) \vee \Big(\pi^{i}_{\mathsf{B}}(v,w) \wedge \pi^{t}_{\mathsf{B}}(v,u,w)\Big)$
	$\rho^{3}_{\mathsf{B}^{\leftrightarrow}}(\{uv,vu\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}^{\leftrightarrow}}(\{vu\},\{\emptyset\},V)$
	$\rho^{4}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{uv,vu\},V)$	$=$	$\rho^{2}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{vu\},V)$
	$\rho^{5}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{vu\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{vu\},V)$
	$\rho^{6}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{uw\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{uw\},V) \vee \Big(v = w \wedge u \neq v\Big)$
	$\rho^{7}_{\mathsf{B}^{\leftrightarrow}}(\{vu\},\{wu\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\leftrightarrow}}(\{vu\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{wu\},V) \vee$ $\Big(v = w \wedge u \neq v\Big) \vee \pi^{t}_{\mathsf{B}}(u,w,v)$
	$\rho^{8}_{\mathsf{B}^{\leftrightarrow}}(\{uv,vw\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}^{\leftrightarrow}}(\{vw\},\{\emptyset\},V)$
	$\rho^{9}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{uv,vw\},V)$	$=$	$\rho^{2}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}^{\leftrightarrow}}(\{\emptyset\},\{vw\},V)$
	$\rho^{10}_{\mathsf{B}^{\leftrightarrow}}(\{uv,wx\},\{wv\},V)$	$=$	$\rho^{7}_{\mathsf{B}^{\leftrightarrow}}(\{uv\},\{wv\},V) \vee \rho^{6}_{\mathsf{B}^{\leftrightarrow}}(\{wx\},\{wv\},V)$
	$\rho^{11}_{\mathsf{B}^{\leftrightarrow}}(\{wv\},\{uv,wx\},V)$	$=$	$\rho^{7}_{\mathsf{B}^{\leftrightarrow}}(\{wv\},\{uv\},V) \vee \rho^{6}_{\mathsf{B}^{\leftrightarrow}}(\{wv\},\{wx\},V)$

Table 8: The expressions to be combined line-by-line as the non-negated form on the left-hand side and negated form on the right-hand side to test for containedness relations.

γ	**Edge Inclusion and Exclusion Implications**		
$(\cdot)^+$	$\rho^{1a}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},\{u,v\})$	$=$	$\pi^e_{\mathsf{B}}(u,v) \wedge \pi^e_{\mathsf{B}}(v,u)$
	$\rho^{1b}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},\{u,v,w\})$	$=$	$\rho^{1a}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},\{u,v\}) \vee \Big(\pi^e_{\mathsf{B}}(u,v) \wedge \Big(\pi^e_{\mathsf{B}}(v,w) \wedge \pi^t_{\mathsf{B}}(v,w,u)\Big)\Big) \vee$ $\Big(\pi^e_{\mathsf{B}}(u,w) \wedge \pi^t_{\mathsf{B}}(u,w,v) \wedge \Big(\pi^e_{\mathsf{B}}(v,u) \vee \big(\pi^e_{\mathsf{B}}(v,w) \wedge \pi^t_{\mathsf{B}}(v,w,u)\big)\Big)\Big)$
	$\rho^{1c}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},\{u,v,w,x\})$	$=$	$\rho^{1b}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},\{u,v,w\}) \vee \rho^{1b}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},\{u,v,x\}) \vee$ $\Big(\Big(\pi^e_{\mathsf{B}}(u,w) \wedge \pi^t_{\mathsf{B}}(u,w,v)\Big) \wedge \Big(\pi^e_{\mathsf{B}}(v,x) \wedge \pi^t_{\mathsf{B}}(v,x,u)\Big)\Big)$
	$\rho^{2a}_{\mathsf{B}^+}(\{\emptyset\},\{uv\},\{u,v\})$	$=$	$\pi^i_{\mathsf{B}}(u,v) \vee \pi^i_{\mathsf{B}}(v,u)$
	$\rho^{2b}_{\mathsf{B}^+}(\{\emptyset\},\{uv\},\{u,v,w\})$	$=$	$\rho^{2a}_{\mathsf{B}^+}(\{\emptyset\},\{uv\},\{u,v\}) \vee$ $\Big(\pi^i_{\mathsf{B}}(u,w) \wedge \pi^t_{\mathsf{B}}(u,v,w)\Big) \vee \Big(\pi^i_{\mathsf{B}}(v,w) \wedge \pi^t_{\mathsf{B}}(v,u,w)\Big)$
	$\rho^{2c}_{\mathsf{B}^+}(\{\emptyset\},\{uv\},\{u,v,w,x\})$	$=$	$\rho^{2b}_{\mathsf{B}^+}(\{\emptyset\},\{uv\},\{u,v,w\}) \vee \rho^{2b}_{\mathsf{B}^+}(\{\emptyset\},\{uv\},\{u,v,x\})$
	$\rho^{3}_{\mathsf{B}^+}(\{uv,vu\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},V)$
	$\rho^{4}_{\mathsf{B}^+}(\{\emptyset\},\{uv,vu\},V)$	$=$	$\rho^{2}_{\mathsf{B}^+}(\{\emptyset\},\{uv\},V)$
	$\rho^{5}_{\mathsf{B}^+}(\{uv\},\{vu\},V)$	$=$	$u \neq v$
	$\rho^{6a}_{\mathsf{B}^+}(\{uv\},\{uw\},\{u,v,w\})$	$=$	$\rho^{1b}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},V) \vee \rho^{2b}_{\mathsf{B}^+}(\{\emptyset\},\{uw\},V) \vee \Big(v = w \wedge u \neq v\Big) \vee$ $\Big(\pi^t_{\mathsf{B}}(u,w,v) \wedge \Big(\pi^e_{\mathsf{B}}(v,u) \vee \Big(\pi^e_{\mathsf{B}}(v,w) \wedge \pi^t_{\mathsf{B}}(v,w,u)\Big)\Big)\Big)$
	$\rho^{6b}_{\mathsf{B}^+}(\{uv\},\{uw\},\{u,v,w,x\})$	$=$	$\rho^{1c}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},V) \vee \rho^{2c}_{\mathsf{B}^+}(\{\emptyset\},\{uw\},V) \vee$ $\rho^{6a}_{\mathsf{B}^+}(\{uv\},\{uw\},\{u,v,w\}) \vee$ $\Big(\pi^t_{\mathsf{B}}(u,w,v) \wedge \Big(\pi^e_{\mathsf{B}}(v,x) \wedge \pi^t_{\mathsf{B}}(v,x,u)\Big)\Big)$
	$\rho^{7}_{\mathsf{B}^+}(\{vu\},\{wu\},V)$	$=$	$\rho^{6}_{\mathsf{B}^+}(\{uv\},\{uw\},V)$
	$\rho^{8}_{\mathsf{B}^+}(\{uv,vw\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}^+}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}^+}(\{vw\},\{\emptyset\},V)$
	$\rho^{9}_{\mathsf{B}^+}(\{\emptyset\},\{uv,vw\},V)$	$=$	$\rho^{2}_{\mathsf{B}^+}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}^+}(\{\emptyset\},\{vw\},V)$
	$\rho^{10}_{\mathsf{B}^+}(\{uv,wx\},\{wv\},V)$	$=$	$\rho^{7}_{\mathsf{B}^+}(\{uv\},\{wv\},V) \vee \rho^{6}_{\mathsf{B}^+}(\{wx\},\{wv\},V) \vee$
	$\rho^{11}_{\mathsf{B}^+}(\{wv\},\{uv,wx\},V)$	$=$	$\rho^{7}_{\mathsf{B}^+}(\{wv\},\{uv\},V) \vee \rho^{6}_{\mathsf{B}^+}(\{wv\},\{wx\},V) \vee$ $\big(\pi^t_{\mathsf{B}}(v,u,w) \wedge \pi^t_{\mathsf{B}}(w,x,v)\big)$

Table 9: The expressions to be combined line-by-line as the non-negated form on the left side and negated form on the right side to test for containedness relations.

γ	**Edge Inclusion and Exclusion Implications**		
$(\cdot)^-$	$\rho^{1a}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},\{u,v\})$	$=$	$\pi^e_{\mathsf{B}}(u,v) \vee \pi^e_{\mathsf{B}}(v,u)$
	$\rho^{1b}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},\{u,v,w\})$	$=$	$\rho^{1a}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},\{u,v\}) \vee$ $\Big(\Big(\pi^e_{\mathsf{B}}(u,w) \wedge \pi^t_{\mathsf{B}}(u,w,v)\Big) \vee \Big(\pi^e_{\mathsf{B}}(v,x) \wedge \pi^t_{\mathsf{B}}(v,x,u)\Big)\Big)$
	$\rho^{1c}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},\{u,v,w,x\})$	$=$	$\rho^{1b}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},\{u,v,w\}) \vee \rho^{1b}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},\{u,v,x\})$
	$\rho^{2a}_{\mathsf{B}^-}(\{\emptyset\},\{uv\},\{u,v\})$	$=$	$\pi^i_{\mathsf{B}}(u,v) \wedge \pi^i_{\mathsf{B}}(v,u)$
	$\rho^{2b}_{\mathsf{B}^-}(\{\emptyset\},\{uv\},\{u,v,w\})$	$=$	$\rho^{2a}_{\mathsf{B}^-}(\{\emptyset\},\{uv\},\{u,v\}) \vee \Big(\pi^i_{\mathsf{B}}(u,v) \wedge \Big(\pi^i_{\mathsf{B}}(v,w) \wedge \pi^t_{\mathsf{B}}(v,u,w)\Big)\Big) \vee$ $\Big(\pi^i_{\mathsf{B}}(u,w) \wedge \pi^t_{\mathsf{B}}(u,v,w) \wedge \Big(\pi^i_{\mathsf{B}}(v,u) \vee \big(\pi^i_{\mathsf{B}}(v,w) \wedge \pi^t_{\mathsf{B}}(v,u,w)\big)\Big)\Big)$
	$\rho^{2c}_{\mathsf{B}^-}(\{\emptyset\},\{uv\},\{u,v,w,x\})$	$=$	$\rho^{2b}_{\mathsf{B}^-}(\{\emptyset\},\{uv\},\{u,v,w\}) \vee \rho^{2b}_{\mathsf{B}^-}(\{\emptyset\},\{uv\},\{u,v,x\}) \vee$ $\Big(\Big(\pi^i_{\mathsf{B}}(u,x) \wedge \pi^t_{\mathsf{B}}(u,v,x)\Big) \wedge \Big(\pi^i_{\mathsf{B}}(v,x) \wedge \pi^t_{\mathsf{B}}(v,u,x)\Big)\Big)$
	$\rho^{3}_{\mathsf{B}^-}(\{uv,vu\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},V)$
	$\rho^{4}_{\mathsf{B}^-}(\{\emptyset\},\{uv,vu\},V)$	$=$	$\rho^{2}_{\mathsf{B}^-}(\{\emptyset\},\{uv\},V)$
	$\rho^{5}_{\mathsf{B}^-}(\{uv\},\{vu\},V)$	$=$	$u \neq v$
	$\rho^{6a}_{\mathsf{B}^-}(\{uv\},\{uw\},\{u,v,w\})$	$=$	$\rho^{1b}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},V) \vee \rho^{2b}_{\mathsf{B}^-}(\{\emptyset\},\{uw\},V) \vee \big(v = w \wedge u \neq v\big) \vee$ $\Big(\pi^t_{\mathsf{B}}(u,w,v) \wedge \Big(\pi^i_{\mathsf{B}}(w,u) \vee \Big(\pi^i_{\mathsf{B}}(w,v) \wedge \pi^t_{\mathsf{B}}(w,v,u)\Big)\Big)\Big)$
	$\rho^{6b}_{\mathsf{B}^-}(\{uv\},\{uw\},\{u,v,w,x\})$	$=$	$\rho^{1c}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},V) \vee \rho^{2c}_{\mathsf{B}^-}(\{\emptyset\},\{uw\},V) \vee$ $\rho^{6a}_{\mathsf{B}^-}(\{uv\},\{uw\},\{u,v,w\}) \vee$ $\Big(\pi^t_{\mathsf{B}}(u,w,v) \wedge \Big(\pi^i_{\mathsf{B}}(w,x) \wedge \pi^t_{\mathsf{B}}(w,x,u)\Big)\Big)$
	$\rho^{7}_{\mathsf{B}^-}(\{vu\},\{wu\},V)$	$=$	$\rho^{6}_{\mathsf{B}^-}(\{uv\},\{uw\},V)$
	$\rho^{8}_{\mathsf{B}^-}(\{uv,vw\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}^-}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}^-}(\{vw\},\{\emptyset\},V)$
	$\rho^{9}_{\mathsf{B}^-}(\{\emptyset\},\{uv,vw\},V)$	$=$	$\rho^{2}_{\mathsf{B}^-}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}^-}(\{\emptyset\},\{vw\},V)$
	$\rho^{10}_{\mathsf{B}^-}(\{uv,wx\},\{wv\},V)$	$=$	$\rho^{7}_{\mathsf{B}^-}(\{uv\},\{wv\},V) \vee \rho^{6}_{\mathsf{B}^-}(\{wx\},\{wv\},V) \vee$ $(\pi^t_{\mathsf{B}}(v,w,u) \wedge \pi^t_{\mathsf{B}}(w,v,x))$
	$\rho^{11}_{\mathsf{B}^-}(\{wv\},\{uv,wx\},V)$	$=$	$\rho^{7}_{\mathsf{B}^-}(\{wv\},\{uv\},V) \vee \rho^{6}_{\mathsf{B}^-}(\{wv\},\{wx\},V)$

Table 10: The expressions to be combined line-by-line as a non-negated form on the left side and negated form on the right side to test for containedness relations.

γ	**Edge Inclusion and Exclusion Implications**		
$(\cdot)^{\rightarrow}$	$\rho^{1}_{\mathsf{B}\rightarrow}(\{uv\},\{\emptyset\},V)$	$=$	$\pi^{e}_{\mathsf{B}}(u,v) \vee \pi^{i}_{\mathsf{B}}(v,u)$
	$\rho^{2}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{uv\},V)$	$=$	false
	$\rho^{3}_{\mathsf{B}\rightarrow}(\{uv,vu\},\{\emptyset\},V)$	$=$	true
	$\rho^{4}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{uv,vu\},V)$	$=$	$\rho^{2}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{vu\},V)$
	$\rho^{5}_{\mathsf{B}\rightarrow}(\{uv\},\{vu\},V)$	$=$	$\rho^{1}_{\mathsf{B}\rightarrow}(\{uv\},\{\emptyset\},V)$
	$\rho^{6}_{\mathsf{B}\rightarrow}(\{uv\},\{uw\},V)$	$=$	$\rho^{1}_{\mathsf{B}\rightarrow}(\{uv\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{uw\},V) \vee v = w$
	$\rho^{7}_{\mathsf{B}\rightarrow}(\{vu\},\{wu\},V)$	$=$	$\rho^{1}_{\mathsf{B}\rightarrow}(\{vu\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{wu\},V) \vee v = w$
	$\rho^{8}_{\mathsf{B}\rightarrow}(\{uv,vw\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}\rightarrow}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}\rightarrow}(\{vw\},\{\emptyset\},V) \vee \pi^{t}_{\mathsf{B}}(v,u,w) \vee u = w$
	$\rho^{9}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{uv,vw\},V)$	$=$	$\rho^{2}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}\rightarrow}(\{\emptyset\},\{vw\},V)$
	$\rho^{10}_{\mathsf{B}\rightarrow}(\{uv,wx\},\{wv\},V)$	$=$	$\rho^{7}_{\mathsf{B}\rightarrow}(\{uv\},\{wv\},V) \vee \rho^{6}_{\mathsf{B}\rightarrow}(\{wx\},\{wv\},V) \vee$ $\Big(\pi^{t}_{\mathsf{B}}(v,u,w) \wedge \pi^{t}_{\mathsf{B}}(w,v,x)\Big) \vee u = w \vee v = x$
	$\rho^{11}_{\mathsf{B}\rightarrow}(\{wv\},\{uv,wx\},V)$	$=$	$\rho^{7}_{\mathsf{B}\rightarrow}(\{wv\},\{uv\},V) \vee \rho^{6}_{\mathsf{B}\rightarrow}(\{wv\},\{wx\},V) \vee u = w \vee v = x$ $\vee\Big(u = x \wedge \pi^{t}_{\mathsf{B}}(w,u,v) \wedge \pi^{t}_{\mathsf{B}}(v,w,u) \wedge \pi^{t}_{\mathsf{B}}(u,w,v)\Big)$
$(\cdot)^{\leftarrow}$	$\rho^{1}_{\mathsf{B}\leftarrow}(\{uv\},\{\emptyset\},V)$	$=$	$\pi^{i}_{\mathsf{B}}(u,v) \vee \pi^{e}_{\mathsf{B}}(v,u)$
	$\rho^{2}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{uv\},V)$	$=$	false
	$\rho^{3}_{\mathsf{B}\leftarrow}(\{uv,vu\},\{\emptyset\},V)$	$=$	true
	$\rho^{4}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{uv,vu\},V)$	$=$	$\rho^{2}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{vu\},V)$
	$\rho^{5}_{\mathsf{B}\leftarrow}(\{uv\},\{vu\},V)$	$=$	$\rho^{1}_{\mathsf{B}\leftarrow}(\{uv\},\{\emptyset\},V)$
	$\rho^{6}_{\mathsf{B}\leftarrow}(\{uv\},\{uw\},V)$	$=$	$\rho^{1}_{\mathsf{B}\leftarrow}(\{uv\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{uw\},V) \vee v = w$
	$\rho^{7}_{\mathsf{B}\leftarrow}(\{vu\},\{wu\},V)$	$=$	$\rho^{1}_{\mathsf{B}\leftarrow}(\{vu\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{wu\},V) \vee v = w$
	$\rho^{8}_{\mathsf{B}\leftarrow}(\{uv,vw\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}\leftarrow}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}\leftarrow}(\{vw\},\{\emptyset\},V) \vee \pi^{t}_{\mathsf{B}}(v,w,u) \vee u = w$
	$\rho^{9}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{uv,vw\},V)$	$=$	$\rho^{2}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}\leftarrow}(\{\emptyset\},\{vw\},V)$
	$\rho^{10}_{\mathsf{B}\leftarrow}(\{uv,wx\},\{wv\},V)$	$=$	$\rho^{7}_{\mathsf{B}\leftarrow}(\{uv\},\{wv\},V) \vee \rho^{6}_{\mathsf{B}\leftarrow}(\{wx\},\{wv\},V) \vee$ $\Big(\pi^{t}_{\mathsf{B}}(v,w,u) \wedge \pi^{t}_{\mathsf{B}}(w,x,v)\Big) \vee u = w \vee v = x$
	$\rho^{11}_{\mathsf{B}\leftarrow}(\{wv\},\{uv,wx\},V)$	$=$	$\rho^{7}_{\mathsf{B}\leftarrow}(\{wv\},\{uv\},V) \vee \rho^{6}_{\mathsf{B}\leftarrow}(\{wv\},\{wx\},V) \vee u = w \vee v = x$ $\vee\Big(u = x \wedge \pi^{t}_{\mathsf{B}}(w,v,u) \wedge \pi^{t}_{\mathsf{B}}(v,u,w) \wedge \pi^{t}_{\mathsf{B}}(u,v,w)\Big)$

Table 11: The expressions to be combined line by line as non-negated form on the left side and negated form on the right side to test for containedness relations.

γ	**Edge Inclusion and Exclusion Implications**		
$(\cdot)^{\oplus}$	$\rho^{1}_{\mathsf{B}^{\oplus}}(\{uv\},\{\emptyset\},V)$	$=$	$\pi^{e}_{\mathsf{B}}(u,v) \vee \pi^{i}_{\mathsf{B}}(u,v) \vee u = v$
	$\rho^{2}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{uv\},V)$	$=$	false
	$\rho^{3}_{\mathsf{B}^{\oplus}}(\{uv,vu\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\oplus}}(\{uv\},\{\emptyset\},V)$
	$\rho^{4}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{uv,vu\},V)$	$=$	$\rho^{2}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{uv\},V)$
	$\rho^{5}_{\mathsf{B}^{\oplus}}(\{uv\},\{vu\},V)$	$=$	true
	$\rho^{6}_{\mathsf{B}^{\oplus}}(\{uv\},\{uw\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\oplus}}(\{uv\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{uw\},V) \vee v = w$
	$\rho^{7}_{\mathsf{B}^{\oplus}}(\{vu\},\{wu\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\oplus}}(\{vu\},\{\emptyset\},V) \vee \rho^{2}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{wu\},V) \vee v = w$
	$\rho^{8}_{\mathsf{B}^{\oplus}}(\{uv,vw\},\{\emptyset\},V)$	$=$	$\rho^{1}_{\mathsf{B}^{\oplus}}(\{uv\},\{\emptyset\},V) \vee \rho^{1}_{\mathsf{B}^{\oplus}}(\{vw\},\{\emptyset\},V)$
	$\rho^{9}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{uv,vw\},V)$	$=$	$\rho^{2}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{uv\},V) \vee \rho^{2}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{vw\},V)$
	$\rho^{10}_{\mathsf{B}^{\oplus}}(\{uv,wx\},\{wv\},V)$	$=$	$\rho^{7}_{\mathsf{B}^{\oplus}}(\{uv\},\{wv\},V) \vee \rho^{6}_{\mathsf{B}^{\oplus}}(\{wx\},\{wv\},V) \vee u = w \vee v = x$
	$\rho^{11}_{\mathsf{B}^{\oplus}}(\{wv\},\{uv,wx\},V)$	$=$	$\rho^{7}_{\mathsf{B}^{\oplus}}(\{wv\},\{uv\},V) \vee \rho^{6}_{\mathsf{B}^{\oplus}}(\{wv\},\{wx\},V) \vee u = w \vee v = x$
	$\rho^{12}_{\mathsf{B}^{\oplus}}(\{vw,wu,uv,$	$=$	$\rho^{3}_{\mathsf{B}^{\oplus}}(\{vw,wv\},\{\emptyset\},V) \vee$
	$wv,vx,xw\},\{\emptyset\},V)$		$\rho^{8}_{\mathsf{B}^{\oplus}}(\{vw,wu\},\{\emptyset\},V) \vee \rho^{8}_{\mathsf{B}^{\oplus}}(\{wu,uv\},\{\emptyset\},V) \vee$
			$\rho^{8}_{\mathsf{B}^{\oplus}}(\{uv,vw\},\{\emptyset\},V) \vee \rho^{8}_{\mathsf{B}^{\oplus}}(\{wv,vx\},\{\emptyset\},V) \vee$
			$\rho^{8}_{\mathsf{B}^{\oplus}}(\{vx,xw\},\{\emptyset\},V) \vee \rho^{8}_{\mathsf{B}^{\oplus}}(\{xw,wv\},\{\emptyset\},V) \vee$
			$\Big(\pi^{t}_{\mathsf{B}}(v,w,u) \wedge \pi^{t}_{\mathsf{B}}(u,w,v) \wedge \pi^{t}_{\mathsf{B}}(w,u,v) \wedge$
			$\pi^{t}_{\mathsf{B}}(v,x,w) \wedge \pi^{t}_{\mathsf{B}}(x,v,w) \wedge \pi^{t}_{\mathsf{B}}(w,v,x)\Big)$
	$\rho^{13}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{vw,wu,uv,$	$=$	$\rho^{4}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{vw,wv\},V) \vee$
	$wv,vx,xw\},V)$		$\rho^{9}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{vw,wu\},V) \vee \rho^{9}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{wu,uv\},V) \vee$
			$\rho^{9}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{uv,vw\},V) \vee \rho^{9}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{wv,vx\},V) \vee$
			$\rho^{9}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{vx,xw\},V) \vee \rho^{9}_{\mathsf{B}^{\oplus}}(\{\emptyset\},\{xw,wv\},V)$

Table 12: The expressions to be combined line-by-line as a non-negated form on the left side and negated form on the right side to test for containedness relations.

The conditions derived for testing the containedness relations of graph classes were also investigated in [61] with a goal-directed symbol elimination approach. It turned out that for the $(\cdot)^{+}$ and $(\cdot)^{-}$ transformation the same conditions were obtained as with the manual second-order quantifier elimination approach. Investigating the remaining simple transformation with this goal-directed approach is a task for future work. Since the maximum size of a minimum witness graph is unbounded for graph classes obtained by a transformation in Γ_2 or Γ_3, no finite universal axiomatization exists for those graph classes. Therefore, the process of goal-directed symbol elimination will not terminate without further assumptions on the predicates. Finding such assumptions will be considered as a task for future work.

4.4 Example Application

4.4.1 Manual Application of the Derived Concept

First the presented concept is illustrated by manually applying it to the graph classes MinDG, MaxDG and CRG and intersections of them.

The inclusion predicate used in the axiom defining $\mathsf{MinDG}(r)$ and the exclusion predicate used in the axiom defining $\mathsf{MaxDG}(\mathsf{r})$ are obviously symmetric, as long as $d(u, v)$ is symmetric (which is assumed for this example). The class $\mathsf{QUDG}(r)$ is defined by axioms using an inclusion and an exclusion predicate, but not by axiom using a transfer predicate, i.e., the transfer predicate is always false. Thus, for testing $(\mathsf{QUDG}(r))^- \not\subseteq (\mathsf{QUDG}(r))^+$ and $(\mathsf{QUDG}(r))^+ \not\subseteq (\mathsf{QUDG}(r))^-$, neither (64) nor (65) of Corollary 15 are satisfied. Therefore, both containedness relations must be satisfied, and in fact both graph classes are equivalent.

Now, the containedness relation between $(\mathsf{DTG}(r))^+$ and $(\mathsf{DTG}(r))^-$ is considered. Applying Corollary 15 together with $\mathsf{DTG}(r) = \mathsf{MaxDG}(r) \cap \mathsf{CRG}$ (i.e. only axioms using exclusion and transfer predicate exists) the satisfiability of

$$\begin{aligned}(\neg d(v, w) > r) \wedge (v \neq u \wedge d(v, u) \leq d(v, w)) \\ \wedge (w \neq x \wedge d(w, x) \leq d(w, v))\end{aligned}$$

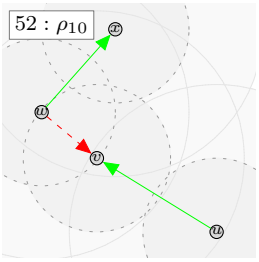
has to be checked for testing $(\mathsf{DTG}(r))^- \not\subseteq (\mathsf{DTG}(r))^+$ and the satisfiability of

$$\begin{aligned}(\neg d(v, u) > r) \wedge (\neg d(w, x) > r) \\ \wedge (v \neq w \wedge d(v, w) \leq d(v, u)) \\ \wedge (w \neq v \wedge d(w, v) \leq d(w, x))\end{aligned}$$

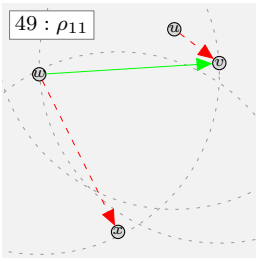
for testing $(\mathsf{DTG}(r))^+ \not\subseteq (\mathsf{DTG}(r))^-$.

This yields the counterexample $d(v, u)$, $d(w, x) \leq d(v, w) \leq r$, $u \neq v$, $w \neq x$, $u \neq w$ and $v \neq x$, which proves $(\mathsf{DTG}(r))^- \not\subseteq (\mathsf{DTG}(r))^+$. As well, this yields the counterexample $d(v, w) \leq d(v, u)$, $d(w, x) \leq r$, $v \neq w$, $u \neq w$ and $v \neq x$, which proves $(\mathsf{DTG}(r))^+ \not\subseteq (\mathsf{DTG}(r))^-$. Thus, neither $(\mathsf{DTG}(r))^+$ nor $(\mathsf{DTG}(r))^-$ is contained in the other class. The according counterexamples are illustrated for $r = 1$ in Fig. 14, where the dashed circles have a radius of $1/2$ and the solid circles have a radius of 1.

Now the assumption, that $d(u, v)$ is symmetric is dropped. In this case $d(u, v) \neq d(v, u)$ is possible. Then the inclusion and exclusion predicate are not necessarily symmetric. Corollary 15 is not applicable. Theorem 14 has to be used to test the containedness relation. The relation between $(\mathsf{QUDG}(r))^-$ and $(\mathsf{QUDG}(r))^+$ is now considered again. For testing the relation $(\mathsf{QUDG}(r))^- \not\subseteq (\mathsf{QUDG}(r))^+$, the second conjunction of expression (63) yields $\neg\rho^2_{\mathsf{A}^-}(\{\emptyset\}, \{uv\}) \wedge \rho^2_{\mathsf{A}^+}(\{\emptyset\}, \{uv\})$. Using the according expression from Table 8 yields $\big(\neg\pi^i_{\mathsf{A}}(u, v) \vee \neg\pi^i_{\mathsf{A}}(v, u)\big) \wedge \big(\pi^i_{\mathsf{A}}(u, v) \vee \pi^i_{\mathsf{A}}(v, u)\big) = \big(\neg\pi^i_{\mathsf{A}}(u, v) \wedge \pi^i_{\mathsf{A}}(v, u)\big) \vee \big(\pi^i_{\mathsf{A}}(u, v) \wedge \neg\pi^i_{\mathsf{A}}(v, u)\big)$.



(**a**) Counterexample proving $(\mathsf{DTG}(r))^+ \not\subseteq (\mathsf{DTG}(r))^-$



(**b**) Counterexample proving $(\mathsf{DTG}(r))^- \not\subseteq (\mathsf{DTG}(r))^+$

Figure 14: Counterexamples for the containedness relations between $(\mathsf{DTG}(r))^+$ and $(\mathsf{DTG}(r))^-$

This formula is satisfied if $d(u,v) < r$ and $d(v,u) \geq r$ or $d(u,v) \geq r$ and $d(v,u) < r$. Thus, a counter example is found that the relation $(\mathsf{QUDG}(r))^- = (\mathsf{QUDG}(r))^+$ does not necessarily hold when $d(u,v)$ is not symmetric.

4.4.2 Automatically Derived Graph Class Containedness Relations

The containedness relations between geometric graph classes resulting from wireless networks, such as QUDG and DTG, under applying simple transformations, i.e. transformations in $\Gamma_1 \cup \Gamma_2 \cup \Gamma_3$, are now derived automatically by using Mathematica for testing the satisfiability of the ρ^i expression in Tables 8 -12. The code used for the test can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main (file Code Section4.nb). The tables in Appendix 9.1 show the automatically derived relations. Since in this specific case the graph class axioms are defined by symmetric and simplified predicates, also containedness relations involving graph classes obtained by simple transformation, that are not in Γ_1 can be checked.

In case that a containedness relation does not hold, a counter example of a witness pattern with at most four vertices can be found. This witness pattern can then be extended to a witness graph, by systematically checking all possible graphs that contain the witness pattern, until a concrete witness graph is found. Since the computational effort required to determine a specific witness graph is significantly higher than that for merely searching for a witness pattern, the analysis in this application scenario is limited to witness patterns. The counter examples constructed in this way are listed in Appendix 9.2. The counter examples are consecutively numbered and labeled with the expression ρ^i (as defined by Tables 8-12 and the negated forms) which led to unsatisfiability of the corresponding containedness relation.

The tables in Appendix 9.1 are to be read from left to right as follows. Class in row i is related to class in column j according to table entry (i,j), by the following relations:

$=$ The classes are the same

$\subset$ The class is contained in the other class but the classes are not the same

$\supset$ The class contains the other class but the classes are not the same

$\times$ Neither of the two classes is contained in the other one.

Putting all the tables together into one single table (with the same class ordering for rows and columns) trivially yields a symmetric matrix with diagonal entries =, however, with reversed relations $\subset$ and $\supset$ for all off-diagonal entries.
By Lemma 2 and Lemma 3 each of the studied graph classes contains at least the null graph. Thus, none of the considered graph classes are disjoint.

The relations $\subset$, $\supset$ and $\times$ in the tables are indexed with the numbered counter examples from Appendix 9.2. Notation $\mathsf{C} \subset_n \mathsf{D}$ refers to the counter example number n, which shows that D is not contained in C. Analogously, $\mathsf{C} \supset_n \mathsf{D}$ refers to the counter example number n, which shows that C is not contained in D. Finally, $\mathsf{C} \times_n^m \mathsf{D}$ refers to the counter example n, which shows that C is not contained in D and the counter example m, which shows that D is not contained in C.

The class name encoding in the tables in Appendix 9.1 uses the definitions of the graph classes from Section 2.3.1. For all graph classes, which are not defined so far a definition is given in the following. An overview about the containedness relations between the graph classes is given in Fig. 15. To avoid clutter, the tables depict the graph classes without the fixed parameters r and 1.

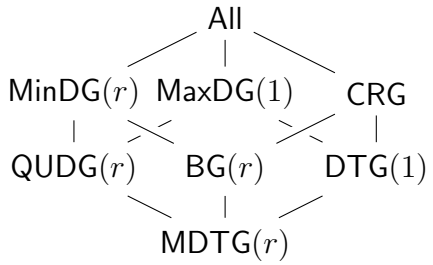


Figure 15: The relations of the base graph classes.

Class All (not defined so far) means all of the predicates (i.e., inclusion, exclusion and transfer) are false. With all predicates being false, all graphs satisfy inclusion, exclusion, and transfer axioms. Thus, the class just contains all graphs over the given vertex universe Ω.

Classes MinDG(r), MaxDG(1) and CRG are as already defined based only on inclusion, exclusion and transfer axioms, respectively. All the remaining classes result from the intersection of the classes, which are predecessors in the diagram (observed from the top to bottom).

The graph class DTG(r) is already defined. The class names BG(r) and MDTG(r) were not used before. They are defined here for the sake of the completeness of the tables. For graphs in class BG(r) (which is termed *boost graphs*), an arbitrary long link can be added to a vertex, which boosts the connectivity of that vertex. It will be connected to all other vertices with the same or less distance. The graphs in class MDTG(r) (which is termed *minimum*

radius DTGs) are directed transmission graphs, but also with a minimum connectivity radius, where vertices are always connected. The graph classes $\mathsf{BG}(r)$ and $\mathsf{MDTG}(r)$ can be built from the base graph classes $\mathsf{MinDG}(r)$, $\mathsf{MaxDG}(1)$ and CRG by

$$\begin{aligned}\mathsf{BG}(r) =& \mathsf{MinDG}(r) \cap \mathsf{CRG}\\ \mathsf{MDTG}(r) =& \mathsf{MinDG}(r) \cap \mathsf{MaxDG}(1) \cap \mathsf{CRG}\end{aligned}$$

Opposed to the usual assumption that $\mathsf{QUDG}(r)$ actually stands for a symmetric subgraph/supergraph of $\mathsf{MinDG}(r) \cap \mathsf{MaxDG}(1)$, it is denoted here just as $\mathsf{MinDG}(r) \cap \mathsf{MaxDG}(1)$ to stay consistent with the notation used in the table. Class $\mathsf{QUDG}(r)$, as usually used in the literature, can be found under QUDG^- and the same class QUDG^+ in the table.

The tables in Appendix 9.1 are tiled into 8×8 blocks, referred as $M(\gamma_1, \gamma_2)$ in the following, each representing the relations of the classes All, MinDG, MaxDG, CRG, QUDG, BG, DTG, and MDTG under the transformations γ_1 and γ_2, i.e., $M(\gamma_1, \gamma_2)$ represents the 64 relations given by

$$\begin{aligned}&\Big(\gamma_1(\mathsf{All}), \gamma_1(\mathsf{MinDG}), \gamma_1(\mathsf{MaxDG}), \gamma_1(\mathsf{CRG}), \gamma_1(\mathsf{QUDG}), \gamma_1(\mathsf{BG}), \gamma_1(\mathsf{DTG}), \gamma_1(\mathsf{MDTG})\Big)\\ \times &\Big(\gamma_2(\mathsf{All}), \gamma_2(\mathsf{MinDG}), \gamma_2(\mathsf{MaxDG}), \gamma_2(\mathsf{CRG}), \gamma_2(\mathsf{QUDG}), \gamma_2(\mathsf{BG}), \gamma_2(\mathsf{DTG}), \gamma_2(\mathsf{MDTG})\Big)\end{aligned}$$

All containedness relations in the tables in Appendix 9.1 were automatically determined by applying `FindInstance` of *Mathematica 14.3* on the derived expression (63) using the Euclidean distance. In connection with the relations of the base classes shown in the Hasse diagram in Figure 15, the following observations can be made for the derived tables resulting from the Mathematica test.

Graph class containedness relations for the same transformation For each $M(\gamma, \gamma)$ block (i.e. when applying the same transformation to the columns and rows of the block), the row for All is the row for MDTG but read reversed from right to left instead of left to right. The same applies for the row pairs ($\mathsf{MinDG}, \mathsf{DTG}$), ($\mathsf{MaxDG}$, BG) and (CRG, QUDG) (which are not directly connected in the Hasse diagram).

Trivially, $M(\bot, \bot)$ and $M(\neg\bot, \neg\bot)$ consists only of entries $=$, since the first always yields the empty graph independent of the source graph, while the latter always yields the complete graph independent of the source graph.

For all other transformations no graph classes collapse into a single class, when the transformation is applied on the base class. Therefore, for all $\gamma \notin \{\bot, \neg\bot\}$ the $M(\gamma, \gamma)$ blocks are equivalent and preserve the relation from the Hasse-Diagramm.

Graph class containedness relations for the same graph class
In the following, observations for the containedness relation $\gamma_1(\mathsf{B}) \subseteq \gamma_2(\mathsf{B})$ are considered. This containedness relation is first investigated for the classes (All, MaxDG, MinDG and CRG).

Afterwards containedness relations for the remaining graph classes are considered.

Case1 (The class All): For the class All the most general classes obtained by transformations are All^{id}, $\mathsf{All}^{\leftrightarrow}$, $\mathsf{All}^{\neg}$ and $\mathsf{All}^{\neg\leftrightarrow}$, since those transformations are neither symmetric, nor oriented, nor complementary oriented and the graphs in those graph classes can therefore contain 0, 1 or 2 of the edges uv and vu. The symmetric transformations, oriented transformations and complementary oriented transformations are all contained in those four general graph classes. However, except for the classes $\mathsf{All}^{\perp}$ and $\mathsf{All}^{\neg\perp}$ none of the graph classes obtained by a symmetric transformations is contained in one of the graph classes obtained by oriented resp. complementary oriented transformations and vice versa. Also none of the graph classes obtained by an oriented transformations is contained in graph class obtained by a complementary oriented transformation and vice versa. Therefore, the graph class relation for all classes between graph classes obtained by oriented and symmetric(without the empty and negated empty transformation), complementry-oriented and symmetric, and as well, oriented and complementry-oriented is always orthogonal, i.e., the 8×8 block $M(\gamma_1, \gamma_2)$ consists only of entries $\times_n^m$ in this case.

Finally $(\mathsf{All})^{\perp}$ is contained in the graph classes obtained by symmetric and oriented transformations and $(\mathsf{All})^{\neg\perp}$ is contained in the graph classes obtained by symmetric and complementary-oriented transformations. Putting all relations together yields the following Hasse diagram.

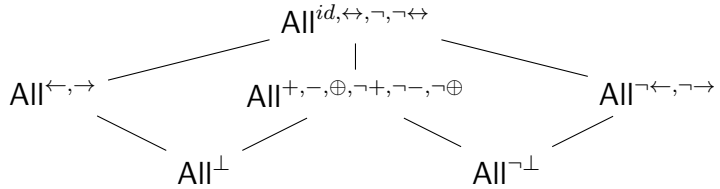


Figure 16: The containedness relations for the class All

Case2 (The class MaxDG): For the class MaxDG a containedness relation can only hold, if it holds in the class All. Furthermore, it has to be distinguished if the exclusion predicate enforces or prohibits the existence of edges in the transformed graph. For all non-negated transformations the exclusion predicate prohibits the existence of edges in the transformed graph, while for all negated transformations the exclusion predicate enforces the existence of edges in the transformed graph. Containedness relations exist only between graph classes obtained by two non-negated or two negated transformations, but not between graph classes obtained by a negated and a by non-negated transformation. The containedness relations that remain between two non-negated resp. two negated transformations are exactly the containedness relation from the class All. Putting all relations together yields the following Hasse diagram.

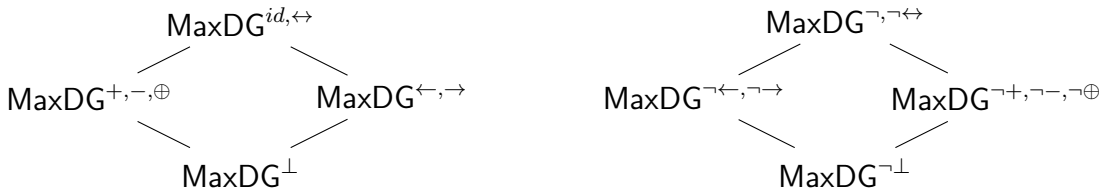


Figure 17: The containedness relations for the class MaxDG

Case3 (The class MinDG): Also for the class MinDG a containedness relation can only hold, if it holds in the class All. As with the testing for the subclass relations of MaxDG it has to be distinguished if the inclusion predicate enforces or prohibits the existence of edges in the transformed graph. The existence of edges is enforced by the inclusion predicate for the classes MinDG^{id}, MinDG^{+}, MinDG^{-}, $\mathsf{MinDG}^{\leftrightarrow}$, $\mathsf{MinDG}^{\neg\rightarrow}$, $\mathsf{MinDG}^{\neg\leftarrow}$, $\mathsf{MinDG}^{\neg\oplus}$ and $\mathsf{MinDG}^{\neg\perp}$. For the remaining transformations the existence of edges is prohibited by the inclusion predicate. Containedness relations exist only between two graph classes obtained by transformations, where the inclusion predicate enforces the existence of edges in the transformed graph or between two graph classes, where the inclusion predicate prohibits the existence of edges in the transformed graph, but not between two graph classes where the inclusion predicate enforces the existence of edges for one transformed graph class and prohibits the existence of edges for the other transformed graph class. The containedness relations that remain between two transformations are exactly the containedness relation from the class All. Putting all relations together yields the following Hasse diagram.

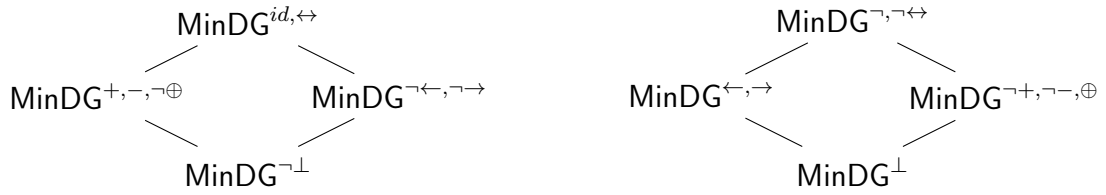


Figure 18: The containedness relations for the class MinDG

Case4 (The class CRG): For the class CRG only a few containedness relations hold. Those are the relations for the classes $\mathsf{CRG}^{\perp}$ and $\mathsf{CRG}^{\neg\perp}$ following from the containedness relations for All with the following two exceptions: $\mathsf{CRG}^{\oplus}$ does not contain $\mathsf{CRG}^{\neg\perp}$ and $\mathsf{CRG}^{\neg\oplus}$ does not contain $\mathsf{CRG}^{\perp}$. Putting all relations together yields the following Hasse diagram.

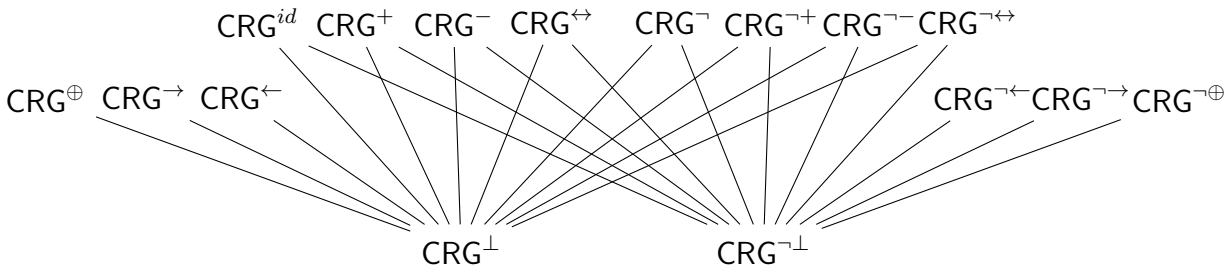


Figure 19: The containedness relations for the class CRG

Case5-8 (The classes QUDG, BG, DTG, MDTG): Analyzing the containedness relations obtained by the Mathemica test for the graph classes (QUDG, BG, DTG and MDTG); it turns out that they can be derived by combining the containedness relations for the classes (All, MaxDG, MinDG and CRG). This means for $\mathsf{C} = \mathsf{A} \cap \mathsf{B}$ and $\gamma_1, \gamma_2 \notin \{\neg\perp, \perp\}$ it holds that $\gamma_1(\mathsf{C}) \subseteq \gamma_2(\mathsf{C})$ iff $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{A})$ and $\gamma_1(\mathsf{B}) \subseteq \gamma_2(\mathsf{B})$. The reason for this equivalence is, that under Assumption 4 and 5, all expressions ρ^i (as defined by Tables 8-12) are given by a conjunctive normal form, where every disjunction contains only inclusion, only exclusion or only transfer predicates.

General graph class containedness relations: The containedness relations between two graph classes are transitive. Therefore, in general for A and B in {All, MinDG, MaxDG, CRG, QUDG, BG, DTG, MDTG} the relation $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{B})$ holds if $\gamma_1(\mathsf{A}) \subseteq \gamma_1(\mathsf{B})$ and $\gamma_1(\mathsf{B}) \subseteq \gamma_2(\mathsf{B})$, such that $\gamma_1(\mathsf{A}) \subseteq \gamma_1(\mathsf{B}) \subseteq \gamma_2(\mathsf{B})$. For the containedness relations obtained by the Mathematica test it turns out that the also $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{B})$ implies $\gamma_1(\mathsf{A}) \subseteq \gamma_1(\mathsf{B})$ and $\gamma_1(\mathsf{B}) \subseteq \gamma_2(\mathsf{B})$. Therefore a complete Hasse diagramm for all containedness relations $\gamma_1(\mathsf{A}) \subseteq \gamma_2(\mathsf{B})$, can be obtained by putting together the Hasse diagramms of the already seperately investigated containedness relations of the form $\gamma_1(\mathsf{A}) \subseteq \gamma_1(\mathsf{B})$ and $\gamma_1(\mathsf{B}) \subseteq \gamma_2(\mathsf{B})$. However such a complete Hasse diagramm can hardly be illustrated clearly anymore and is therefore omitted.

5 Construction of Connected and Plane Subgraphs

Constructing a plane subgraph, by choosing a subset of edges from a given finite graph, while preserving the connectivity of the graph is only possible for graphs that have some structural properties. In order to obtain such a connected and plane subgraph, it has to be guaranteed that for every removed edge an alternative path exists, that connects the vertices of the removed edge. Checking if such a path exists requires global knowledge on the graph, since such a path can be arbitrary long. Therefore, an even more challenging task is to construct such a connected and plane subgraph by local information only.

One property that is investigated in the context of constructing connected plane subgraphs, is the redundancy property, where for every pair of intersecting edges at least one of the four vertices involved in the intersection is connected by an edge to the vertices of the other edge. However, this property alone is not sufficient for the existence of a connected and plane subgraph. Therefore, additional properties are required for the construction of a connected and plane subgraph. One possibility is to consider a stricter version of the redundancy property, the acyclic redundancy, where the redundancy edges have to be shorter than the redundant edge. Another possibility is to consider in addition to the redundancy property, the coexistence property, where vertices located inside a triangle formed by three edges are connected by an edge to the border vertices of a triangle.

In this chapter, first graphs satisfying the acyclic redundancy property are investigated. A local algorithm for the construction of connected and plane subgraphs is presented. Afterwards, it is proven that also for all graphs in the class RCG a connected and plane subgraph exists. Finally, for the graphs in RCG the local CP-algorithm [3] for the construction of connected and plane subgraphs is presented. Since the CP-algorithm was not proven to be correct so far, a correctness proof of the CP-algorithm is given in Section 7. Even if the correctness proof of the CP-algorithm makes the existence proof of a connected and plane spanning subgraph for graphs in RCG irrelevant, the existence proof is nevertheless presented in the following, since some results from the existence proof were helpful for the design and the correctness proof of the CP-algorithm.
Parts of the results presented in this section were already published in [7, 11] and [10].

5.1 Connected Planarization Under Acyclic Redundancy Property

The acyclic redundancy property was studied in [7]. For the acyclic redundancy property not only the existence of the edges, but also their length resp. weight has to considered. The acyclic redundancy property is satisfied, if the redundancy edges are shorter resp. have lower weight than the redundant edge. All properties and algorithms are defined on (injectivity) weighted graphs. Since a strict total edge order can also be transferred to a weight function, everything holds as well for any arbitrary strict total ordering of the edges.

Definition 26 (**Basic acyclic redundancy [7]**). *A weighted embedded graph (G, c) satisfies the basic acyclic redundancy property (BAR) if for each pair of intersecting edges uv, wx one of the intersecting edges is the c-heaviest edge of a triangle in G within $\{u, v, w, x\}$.*

An example with Euclidean distance as weight function is shown in Fig. 21(**a**).

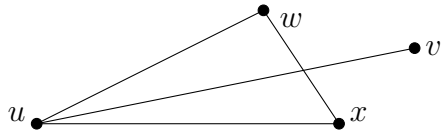
Definition 27 (**Extended acyclic redundancy [7]**). *A weighted embedded graph (G, c) satisfies the* extended acyclic redundancy property (EAR) *if for each pair of intersecting edges uv, wx one of the intersecting edges is the c-heaviest edge of a triangle or quadrilateral in G within $\{u, v, w, x\}$.*

If it is not further specified whether basic or extended acyclic redundancy is regarded, just acyclic redundancy is used. Graphs satisfying the acyclic redundancy are termed as ARG.

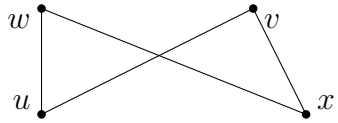
Definition 28 (**Euclidean redundancy property [7]**). *An embedded graph G satisfies the* basic (extended) Euclidean redundancy property *if (G, d_2) satisfies the basic (extended) acyclic redundancy property, where d_2 is the Euclidean distance between two vertices.*

Since basic acyclic redundancy is a stricter version of proper redundancy (PR), PR is implied by BAR. Furthermore, extended acyclic redundancy is implied by basic acyclic redundancy, since EAR is an extension of BAR.

Other relations between EAR, BAR and PR do not hold. Examples of graphs, proving that certain containedness relations between the graph class do not hold are presented in the following. For EAR and PR the examples illustrated in Fig. 20(**a**) and Fig. 20(**b**)) show that neither $\mathsf{EAR} \not\subseteq \mathsf{PR}$, nor $\mathsf{PR} \not\subseteq \mathsf{EAR}$. Furthermore, BAR is not implied by one of the properties EAR or PR, nor of their conjunction (Fig. 21(**b**)). However, BAR implies UDG and therefore also EAR implies UDG. The other implication does not hold (see Fig. 21(**a**)). An overview about all containedness relations, can be found in (see Fig. 22).



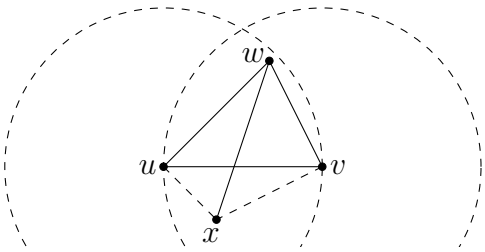
(**a**) Example for an intersection that satisfies the proper redundancy property, but not the extended acyclic redundancy property.



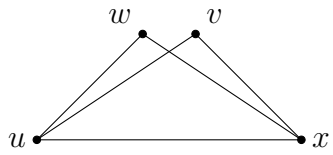
(**b**) Example for an intersection that satisfies the extended acyclic redundancy property, but not the proper redundancy property.

Figure 20: Example of intersections, showing the EAR and PR are not contained in each other.

For graphs satisfying the BAR resp. EAR property a global and a local algorithm for constructing a plane and connected subgraph was developed and proven to be correct in [7]. Since BAR is implied by EAR, only the algorithms for graphs satisfying the EAR property are regarded in the following.



(**a**) Example for a graph that satisfies the Euclidean acyclic redundancy which is not a UDG.



(**b**) Example for an intersection that satisfies proper and extended acyclic redundancy but not basic acyclic redundancy.

Figure 21: Example of intersections, showing that $\mathsf{UDG} \not\subseteq \mathsf{BAR}$ and $\mathsf{BAR} \not\subseteq \mathsf{PR} \cap \mathsf{EAR}$.

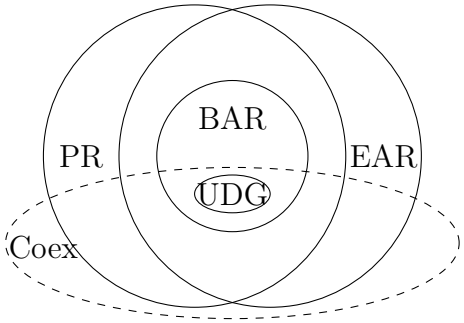


Figure 22: Venn diagram depicturing the relations of the graph classes satisfying the different redundancies, the unit-disk-graph-property and the coexistence property.

Algorithm 1 Global algorithm for extended acyclic redundancy

Input: A finite and connected graph $G = (V, E)$ in the class EAR
Output: A connected and plane spanning subgraph $G' = (V, F)$ of G

```
F ← E (* initialize the result *)
W ← E (* set up the working set *)
while W ≠ ∅ do
    choose c-heaviest uv ∈ W
    remove uv from W
    if ∃uw, vw ∈ F ∨ ∃uw, vx, wx ∈ F then
        remove uv from F
    end if
end while
G' = (V,F)
```

Since in [7] only a correctness proof for the global BAR-algorithm was explicitly given, a short correctness proof of the EAR-algorithm, following the correctness proof of the BAR-algorithm is given here.

Theorem 6. *Algorithm 1 outputs a connectivity-preserving plane subgraph under extended acyclic redundancy.*

Proof. The connectivity follows directly from the formulation of the algorithm, since an edge uv can only be removed if in the so far constructed subgraph there exists still an alternative path of edges uw and wv or of edges uw, wx and vx left.

For the planarity, it is assumed there remains an intersection between the edges uv and wx after the termination of the algorithm. Since uv was not removed, it was not the longest edge of a triangle within the vertices u, v, w or a quadrilateral within the vertices u, v w, x. However, because of the extended acyclic redundancy property, wx was then the longest edge of a triangle within the vertices u, w, x or the longest edge of a quadrilateral within the vertices u, v, w, x. In both cases wx would have been removed. Since this holds for all intersections, the resulting graph G' is intersection free. □

Algorithm 2 Distributed 2-hop algorithm from the view of an vertex u

Input: A finite and connected graph $G = (V, E)$ in the class EAR
Output: A connected and plane spanning subgraph $G' = (V, F)$ of G

$F \leftarrow E$ (* initialize the result *)
for $v \in N_1(u)$ **do**
  if $\exists uw, vw \in E | c(uw), c(vw) < c(uv) \vee \exists uw, vx, wx \in E | c(uw), c(vx), c(wx) < c(uv)$ **then**
    remove e from F
  end if
end for
G' = (V,F)

The correctness of the local EAR-algorithm was proven in [7] by proving the following Theorem.

Theorem 7. *Algorithm 2 outputs the same graph as Algorithm 1.*

In [7] and the subsequent work, [8] and [9] the likelihood of the EAR-property and the connectivity of graphs obtained after applying the EAR algorithm was investigated in terms of simulations. It turned out that the EAR-property is quite likely to hold and the EAR-algorithm is a suitable approach for the connected planarization of graphs, when models related to realistic WSNs are considered.

However, a disadvantage of the EAR-algorithm is that the resulting graphs are quite sparse, since they do not contain any triangle or quadrilaterals anymore. Therefore, the EAR-algorithm removes much more edges from a given edge set, then it would be required to obtain a plane subgraph from a given input graph. In order to obtain plane and connected spanning subgraphs, that preserve more edges from a given input graph, in the following the class of RCG is considered. For graphs in the class RCG an algorithm is developed, that constructs a maximal plane output graph, with respect to the input graph.

5.2 Connected Planarization for RCGs

In order to show that for every finite connected graph $G = (V, E)$ in the class RCG, a connected and plane subgraph $G' = (V, F)$ can be constructed, the approach *Localized Link Removal and Addition based Planarization algorithm (LLRAP)* from [52] for the connected planarization of a graph in the class RCG is considered first.

Algorithm 3 Global version of the LLRAP planarization algorithm

Input: A finite and connected graph $G = (V, E)$ in the class RCG
Output: A connected and plane spanning subgraph $G' = (V, F)$ of G

$G \leftarrow (V, E)$

(* removal phase*)
$F \leftarrow E$ (* set up the result set *)
$W \leftarrow E$ (* set up the working set *)
while $W \neq \emptyset$ **do**
  choose $uv \in W$
  remove uv from W
  if $\exists wx \in E$ intersecting with uv $((uw \wedge vw \in E) \vee (ux \wedge vx \in E))$ **then**
    remove uv from F
  end if
end while

(* addition phase*)
$W \leftarrow E \setminus F$ (* set up the working set for the addition phase*)
while $W \neq \emptyset$ **do**
  choose $uv \in W$
  remove uv from W
  if $\forall wx \in E$ intersecting with uv $(wx \notin F)$ **then**
    add uv to F
  end if
end while

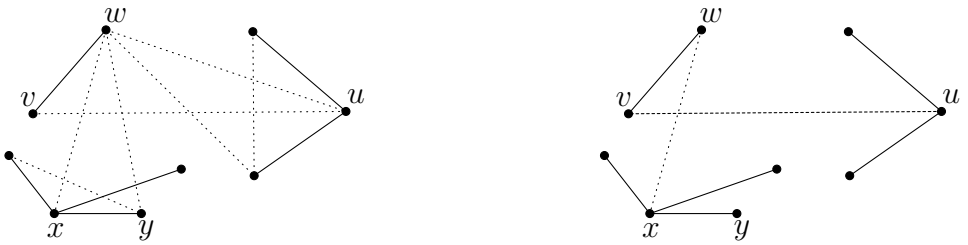
$G' \leftarrow (V, F)$

The LLRAP algorithm (see Alg. 3) always contructes a plane subgraph, but connectivity is not always guarnated. The algorithm is divided into two phases, the *local crosslink detection and removal* phase and the *local link addition* phase. In the first phase all pairs of intersecting edges are detected in the original network graph G and removed according to the following rule: Each edge uv, which is intersected by an edge wx in G is deleted if either both of the edges uw and wv or both of the edges ux and xv exist in G. The redundancy property guarantees that for each pair of intersecting edges uv and wx at least one of the following edge pairs exists: (1) uw and ux, (2) vw and vx, (3) wu and wv or (4) xu and xv (see Fig. 2(**a**)).

Therefore, for each intersection at least one of the intersecting edges is removed. This leads to a plane graph, after the removal phase, but connectivity is not guaranteed anymore.

In order to connect the graph again a subsequent addition phase is contained in the algorithm. In the addition phase an edge uv is added to the graph again, if it does not intersect with an edge of the so far constructed graph G'. This leads to the graph $G'' = G' + uv$ in that step. Since all intersections are evaluated and removed in the removal phase and edges are added in the addition phase only if they are not intersecting with the intermediate result, the final resulting graph is obviously a plane graph. However, in the addition phase there is no specific order for adding edges. Therefore, the case that an added edge blocks all other edges suitable to connect two components may occur (see Fig. 23).

A first approach to solve this problem might be to require that an added edge has to connect two components in the so far constructed graph G'. This might be a plausible approach, because adding an edge that connects two vertices of the same component will only block other possible edges which could be re-added to the graph, but is no improvement on the way to construct a connected graph. However, there are still graphs where LLRAP does not lead to a connected graph, even if the condition, that an edge can only be added if it connects two so far unconnected components of the graph (see Fig. 23) is added. Therefore, there is no easy way to determine the order in which the edges should be re-added to the graph. However, this order is determining whether the algorithm finds a solution or not.



(**a**) Dotted edges are removed in the removal phase of LLRAP

(**b**) If the edge uv is added, wx is blocked and the graph remains disconnected

Figure 23: Example for a graph where LLRAP could fail

In order to understand the problem, why the graph becomes disconnected, the example of Fig. 23 is regarded in detail. The reason why the graph becomes disconnected in this example is that an edge wx is removed, because there exists an alternative path $p(w; y; x)$ containing the edges wy and xy. However, the edge wy is also removed since there exists an alternative path $p(w; x; y)$ containing the edges wx and xy. So the edges wx and wy are part of the alternative path for each other. In this example it is easy to see that the edges connecting the two components depend on each other. However, such a dependency can also contain more than two edges and therefore arbitrary long cycles can occur. To show this the following notation is introduced:

Definition 29 (**Redundancy sequence and redundancy cycle).** *A redundancy sequence S is a sequence of edges $e_1, \ldots, e_k$, where e_i is a redundancy edge for e_{i-1} for $i \in 2, ..., k$. A redundancy cycle R is a redundancy sequence, where e_1 is a redundancy edge for e_k. The edges of a redundancy cycle are termed as cyclic redundant edges.*

The distance $|uv, wx|$ between two edges uv and wx is defined as the length of the shortest path connecting a vertex of the edge uv with a vertex of the edge wx. The following theorem states that such a redundancy cycle can be arbitrary long.

Theorem 8. *For each $n \in \mathbb{N}$, always a graph containing a redundancy cycle R can be constructed, such that for each edge $u_k v_k \in R$ holds, that another edge $u_l v_l \in R$ exists with $|u_k v_k, u_l v_l| > n$.*

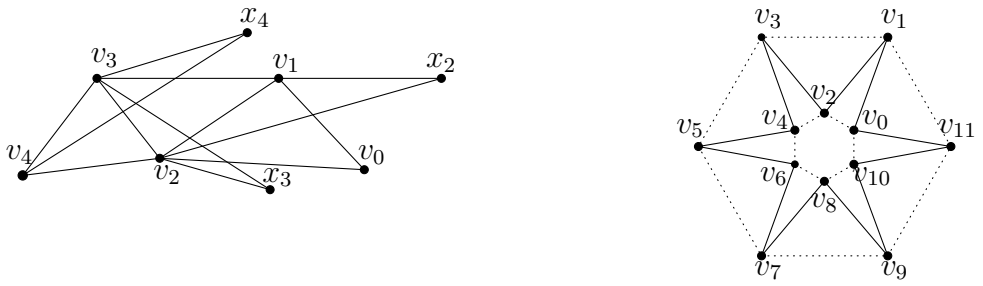
Proof. A graph containing an arbitrarily long redundancy cycle can be constructed by using the following scheme: Two vertices v_0 and v_1 and the edge $v_0 v_1$ are set. After that, a node v_2 and the edges $v_0 v_2$ and $v_1 v_2$ are added to the graph (see Fig. 24).

For $m \in \mathbb{N}$ the graph is constructed in the following way: For every i with $3 \leq i \leq 2m+2$ the vertices v_i and x_i are set, such that $v_i x_i$ only intersects $v_{i-3} v_{i-1}$ and $v_{i-2} v_{i-1}$. Furthermore, for every j with $2 \leq j \leq i$ no vertex is located inside the triangle $\Delta v_{j-2} v_{j-1} v_j$ and $v_{i-1} v_i$ does not intersect $v_{j-2} v_{j-1}$. Afterwards, the edge $v_{i-1} x_i$ is added, such that G satisfies the redundancy property (see Fig. 24 (a)). It also holds that, $v_{2m} = v_0$, $v_{2m+1} = v_1$ and $v_{2m+2} = v_2$.
Each graph which is constructed this way satisfies the redundancy property and

$$R = \{\{v_0, ..., v_{2m-1}\}, \{(v_0, v_1), (v_1, v_2), ..., (v_{2m-2}, v_{2m-1}), (v_{2m-1}, v_0)\}\}$$

is a redundancy cycle with $|E_R| = 2m$. All edges (v_i, v_{i+2}) are not part of R. These edges form two envelopes: An outer envelope and an inner envelope (see Fig. 24 (b)).

Since for all i the vertices v_i, v_{i+1}, v_{i+2} form triangles, the edges $(v_i,\ v_{i+2})$ and (v_{i+1}, v_{i+3}) cannot be part of the same envelope. Hence, all of the edges (v_i, v_{i+2}) with even indices



(**a**) Part of the redundancy cycle used in the proof of Theorem 8

(**b**) Resulting redundancy cycle without the additional intersecting edges

Figure 24: The scheme to construct arbitrary long redundancy cycles

form one envelope, while all of the edges with odd indices form the other one. It follows, that each envelope contains m edges. It is also clear, that the shortest path between two cyclic redundant edges does not contain any other cyclic redundant edge, because every cyclic redundant edge has a vertex on each envelope. Hence, the maximum distance between two cyclic redundant edges is $\lfloor \frac{m}{2} \rfloor$.

Let a graph be constructed as described above with $m \geq 2(n+1)$. Then, for each edge $(v_k, v_{k+1}) \in R$ it holds, that there exists another edge $(u_l, v_{l+1}) \in R$, for $0 \leq k, l \leq m-1$ with $|(v_k, v_{k+1}), (v_l, v_{l+1})| > n$. □

The theorem shows that it is not possible to find out with local information if two edges are part of the same redundancy cycle. If such a redundancy cycle would not occur the LLRAP algorithm would work correctly, since for each deleted edge an alternative path would still exists in the graph. The task is now to handle the redundancy cycles and find a way to construct a connected graph.

In order to find such a connected and plane subgraph of a graph in the class of RCG some properties have to be proven first. These properties are then used to prove the existence of a connected and plane subgraph.

Lemma 16. *Deleting a vertex v in a RCG yields again an RCG $G' = G \setminus v$.*

Proof. Let $G = (V, E)$ be a RCG and $G' = G \setminus v$ the induced subgraph obtained by deleting a single vertex $v \in V$ from G. For any pair of intersecting edges, not involving the vertex v the redundancy property is satisfied in G' iff it is satisfied in G. Furthermore, any pair of intersecting edges, involving the vertex v does not exist anymore. Therefore the redundancy property is satisfied for all pairs of intersecting edges in G'. Analogously, only triangles not involving the vertex v can exist in G'. Furthermore, since v does not exist in G', it is also not possible that v is located inside a triangle in G'. Therefore, also the coexistence property is satisfied in G'. Consequently, deleting a vertex from a RCG yields again a RCG. □

Lemma 17. *All vertices that are located inside a triangle form a clique.*

Proof. If there are no vertices located inside a triangle Δuvw the statement is obviously true. So it is assumed by induction that the clique property holds for n vertices located inside the triangle. Adding another vertex x yields by coexistence property the existence of the edges ux, vx and wx and therefore the triangles Δuvx, Δuwx and Δwvx have to exist. Since all n vertices in the triangle Δuvw are located inside (at least) one of the triangles Δuvx, Δuwx or Δwvx, edges from x to all n vertices exist. These n vertices together with x as well as u, v and w form again a clique. □

Lemma 18. *Let $G = (V, E)$ be a RCG and let $G' = (V', F)$ be a connected and plane subgraph of $G \setminus v$, where $V' = V \setminus v$ and $F \subset E$. Let uv be an edge in G which intersects with some edges of F. Among the edges in F that intersect with uv, let w_1x_1 be the edge with intersection point closest to u or intersection point closest to v. Then the following holds:*

(1) If w_1x_1 is closest to v and one of the edges vw_1 or vx_1 exist, then G' can be transformed into a connected and plane subgraph $G_1 = (V, F_1)$ of G, where F_1 contains all edges of F and a path $p(v; w_1)$ or $p(v; x_1)$ that does not intersect with an edge of F and not with uv (see Fig. 25 (a)).

(2) If w_1x_1 is closest to u and one of the edges uw_1 or ux_1 exist, Then G' can be transformed into a connected and plane subgraph $G_2 = (V', F_2)$ of $G \setminus v$, where F_2 contains all edges of F and a path $p(u; w_1)$ or $p(u; x_1)$ that does not intersect with an edge of F and not with uv (see Fig. 25 (b) with one of the paths $p(u; w_1)$ or $p(u; x_1)$).

(3) If w_1x_1 is closest to u and both edges uw_1 and ux_1 exist, then G' can be transformed into a connected and plane subgraph $G_3 = (V', F_3)$ of $G \setminus v$ which does not contain the edge w_1x_1 (see Fig. 25 (b)).

(**a**) w_1x_1 is closest to v (**b**) w_1x_1 is closest to u

Figure 25: The situations of Lemma 18

Proof. Let the intersection point of uv and w_1x_1 be p (p is not a vertex).

(1) and (2): The proof is given for the case (1). The proof for case (2) is analogous. It is assumed that the edge w_1x_1 is the edge of F with intersection point closest to v and the edge vw_1 exists. By redundancy property at least one of the edges vx_1 or uw_1 has to exist also. Wlog. this is uw_1 and therefore the triangle Δvw_1u exists. Then the vertices that form the convex hull of the vertices inside the triangle Δvpw_1 are regarded. This leads to the a separation of the triangle into two areas, one area containing all vertices in the triangle and another one that is empty. By Lemma 17 all vertices in the triangle Δvw_1u (and therefore also in Δvpw_1) are connected. Therefore, the path $p(v; w_1)$ on the boundary of the convex hull of the vertices inside the triangle Δvpw_1 exists in G.

This path cannot be intersected by another edge of F since otherwise the edge uv has to be intersected by another edge w_ix_i of F, where the intersection point p_i of the edges uv and w_ix_i is closer to v than p or the edge w_ix_i is intersecting the edge w_1x_1. Both is a contradiction and therefore $F' = F + p(v; w_1)$ satisfies the condition.

(3): The triangle Δuw_1x_1 has to exist and can be split into two triangles Δupw_1 and Δupx_1. Now the vertices that form the convex hull of the vertices inside the triangle Δupw_1

and as well the vertices that form the convex hull of the vertices inside the triangle Δupx_1 are regarded. Each of the triangles is divided in two areas, one containing all vertices in the triangle and another one that is empty. By Lemma 17 all vertices in the triangle Δuw_1x_1 (and therefore also in Δupw_1 and Δupx_1) are connected and therefore the path $p(u; w_1)$ on the boundary of the convex hull of the vertices inside the triangle Δupw_1 and the path $p(u; x_1)$ on the boundary of the convex hull of the vertices inside the triangle Δupx_1 exist in G.

Both paths cannot be intersected by another edge of F since otherwise the edge uv has to be intersected by another edge w_ix_i of F, where the intersection point p_i of the edges uv and w_ix_i is closer to u than p or the edge w_ix_i is intersecting the edge w_1x_1. Both is a contradiction and therefore $F_3 = F \setminus w_1x_1 + p(u; w_1) + p(u; x_1)$ satisfies the condition. □

Lemma 19. *Let $G = (V, E)$ be a RCG and let $G' = (V', F)$ be a connected and plane subgraph of $G \setminus v$, where $V' = V \setminus v$ and $F \subset E$. Let $e_i = w_ix_i$ be a sequence of edges in F. Let vw_i be a sequence of edges, where for all $i \geq 1$ the edge vw_{i-1} is intersected by w_ix_i. Among all edges in F, which are intersecting vw_{i-1}, the edge w_ix_i is choosen such that the intersection point p_i of the edges vw_{i-1} and w_ix_i is closest to w_{i-1}. Furthermore, for the intersection of vw_{i-1} and w_ix_i one of the edges vw_i or vx_i exists. Then the labeling of the edge w_ix_i can be choosen such that the edge vw_i exists and one of the following properties holds for all i and k with $k > i$:*

(1) The edge w_kx_k is intersecting with the edge vw_i.

(2) There exists a j with $k > j > i \geq 0$, such that the vertex w_k is located inside the triangle Δvp_jw_j.

Proof. The proof is given by induction. For $k = i + 1 \geq 1$ the edge w_kx_k is intersecting the edge vw_i and therefore property (1) is satisfied for the induction base. For $k \geq i + 2 \geq 2$ it is assumed that the one of the two properties holds for $k - 1$.

First the case that property (1) is satisfied for $k-1$ is considered. An edge w_kx_k intersecting with vw_i satisfies property (1). Therefore, it can be assumed that w_kx_k is not intersecting with vw_i. In this case one of the vertices w_k or x_k is located inside the triangle $\Delta vp_{k-1}w_{k-1}$, because the edges $w_{k-1}x_{k-1}$ and w_kx_k cannot intersect, since they are both in F. The labeling is chosen such that w_k is the vertex located inside the triangle $\Delta vp_{k-1}w_{k-1}$. Furthermore, by coexistence property the edge vw_k exists, because the edge vw_{k-1} exists and by redundancy property also one of the edges w_iw_{k-1} or vx_{k-1} exists. So property (2) is satisfied in this case.

Now the case that property (2) is satisfied for $k - 1$ is considered. Also in this case for an edge w_kx_k intersecting vw_i property (1) is satisfied. If w_kx_k is not intersecting with vw_i, then one of the vertices w_k or x_k is located inside the triangle Δvp_jw_j and the labeling can be chosen such that w_k is located inside the triangle Δvp_jw_j. Then again the edge vw_k exists by coexistence property, since one of the edges w_iw_j or vx_j has to exist by redundancy property.

Therefore, in all possible cases one of the properties (1) or (2) is satisfied. □

The properties from the previous Lemmas are now used to construct a connected and plane spanning subgraph for a given graph in the class RCG.

Theorem 9. *For each connected graph $G = (V, E)$ in the class* RCG *there exists a spanning subgraph $G' = (V, F)$ such that G' is connected and plane.*

Proof. To find a connected and plane subgraph of a given graph G in RCG a recursive approach by adding vertices to an empty initial solution and restructuring the existing graph with Lemma 18 whenever this is necessary is considered.

To show the correctness of that approach an induction argument is applied. Starting with a single-vertex graph it is clear that the properties are satisfied. As induction hypothesis it is assumed that every graph $G = (V, E)$ with at most n vertices in the class RCG contains a connected and plane spanning subgraph G'. So for an arbitrary connected graph with $n + 1$ vertices in the class RCG a spanning tree is regarded (not necessary a plane graph) and a leaf v of this tree is deleted. This leads to a graph $G \setminus v$, which is still connected.

The graph $G \setminus v$ is by Lemma 16 still contained in the class RCG and by the induction hypothesis there exists a connected and plane graph $G' = (V', F)$ containing all vertices and a subset of edges of $G \setminus v$. If there exists an edge that connects v to a vertex of G' without intersecting an edge in F, then this edge can be added to F and the proof is completed.

So it is assumed that all edges, that connect v with a vertex of G' are intersected by an edge in F. Let one of this edges be uv and let $wx \in F$ be the edge, where the intersection point p of the edges uv and wx is closest to v among all edges in F intersecting with uv (see Fig. 26). By redundancy property one of the four vertices u, v, w, x has to be connected by an edge to the other three vertices. If this vertex is not u, then at least one of the edges vw or vx has to exist, say wlog. vw. By Lemma 18(1) a path $p(v; w)$ on the convex hull of the triangle Δvpw can be found and this path forms together with the edges of F a plane graph containing all vertices of G (see Fig. 26 (a)).

Thus, it can be assumed that only the edges uw and ux exist. If there exists no other edge of F intersecting with uv or for all other intersecting edges, u is connected by an edge to both vertices of the intersecting edges, Lemma 18(3) can be applied multiple times and uv can be added to G'. Therefore, it can be assumed that there exists an intersecting edge w_1x_1 where u is not connected by an edge with both vertices of this edge. The labeling is chosen such that w_1 is located on the same side of uv as w and x_1 is located on the same side of uv as x.

Now the edge $w_1x_1 \in F$, intersecting with uv, with intersection point p_1 closest to u, where u is not connected by an edge to both vertices of the intersecting edge is considered (see Fig. 26 (b)). For all previous intersections Lemma 18(3) is applied. Now two cases have to be distinguished. In the first case v is connected by an edge with both vertices of w_1x_1. If w or x are located inside the triangle Δw_1x_1v there has to exist one of the edges vw or vx by coexistence property, in contradiction to the assumption. In the other case the edges w_1v and wu as well as x_1v and xu are intersecting. Then by redundancy property either one of the

edges vw or vx has to exist or both of the edges w_1u and x_1u. This is also a contradiction to the assumption (see Fig. 26 (b)).

(**a**) The vertex v can be added to $G \setminus v$ by a path $p(v; w)$

(**b**) u is connected by an edge with w and x as well as v with w_1 and x_1

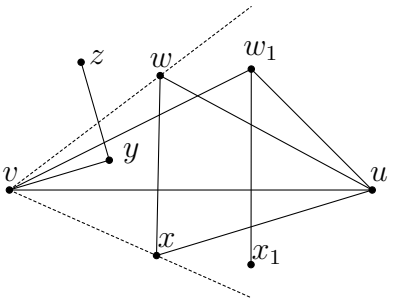
Figure 26: The easy cases when u or v are connected to the intersecting edges

Thus, the only remaining case is, that neither u nor v are connected by an edge with w_1 and x_1 (see Fig. 27). Then w_1 or x_1 have to be connected by an edge with u and v, say wlog. w_1. Since w_1x_1 with intersection point p_1, was the first (remaining) intersecting edge starting from u, by Lemma 18(2) a path $p(u; w_1)$ on the convex hull of the triangle Δup_1w_1 exists that is not intersected by an edge in F. However, it is not sure that such a path can also be constructed in the triangle Δw_1p_1v since there can be intersecting edges in F.

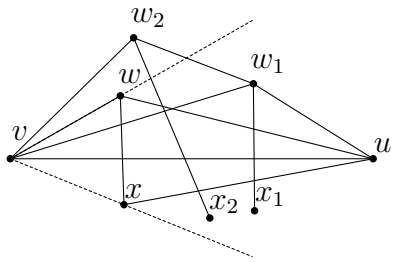
Next, the intersecting edges in F are considered for the edge vw_1. Let q be the intersection point of vw_1 and wx. If the edge in F with intersection point closest to v among all edges in F intersecting vw_1 is not wx, but an edge yz, then one of the vertices y or z is located inside the triangle Δvqw (see Fig. 27 (a)) and Lemma 18(1) can be applied to connect v to y resp. z. So wx can be assumed to be the edge in F with intersection point closest to v among all edges in F intersecting with vw_1. Then for the edge vw_1 the edge in F with intersection point closest to w_1 is considered. As for the edge uv, Lemma 18(3) is applied to remove all intersections, until an edge w_2x_2 is reached, where only the redundancy edges vw_2 and w_1w_2 exist. This edge is then considered in the same way as uv and vw_1. Repeating this process yields a sequence S of edges vw_0, vw_1,...,vw_i, where $u = w_0$.

It remains to prove that all edges vw_i of this sequence intersect with wx. If there would exist an edge vw_k that does not intersect with wx but all previous edges of the sequence did, then one of the vertices w or x would be located inside the triangle Δvw_kw_{k-1} and therefore by coexistence property the edge vw or vx would exists, in contrast to the assumption (see Fig. 27 (b)). Therefore, the vertex that is connected by an edge with v has to be located in the cone C_1 formed by v, w and x, but outside the (virtual) triangle Δvwx. Consequently all edges vw_i can be assumed to intersect with wx.

From Lemma 19 it follows that the sequence S cannot be cyclic. Since the graph G is finite there has to exist an edge vw_n, which can be added to G' after applying Lemma 18(3) for all intersecting edges in F (see Fig.28). □

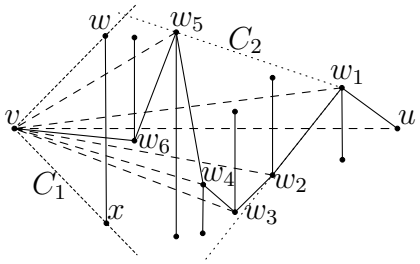


(**a**) Edge wx is not the first intersecting edge with vw_1 starting from v

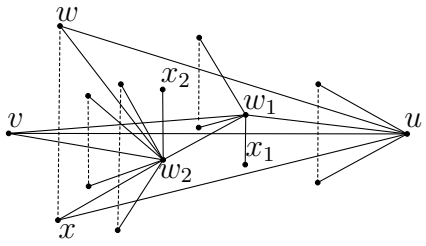


(**b**) Vertex w is located inside a triangle

Figure 27: Two situations of sequences of edges vw_k



(**a**) Sequence of edges vw_k and cones C_1 and C_2 (where u is w_0)



(**b**) Edge vw_2 can be added to F after restructuring the graph with Lemma 18

Figure 28: Possible sequences of edges vw_k to illustrate the termination

5.3 The Connected Planarization Algorithm for RCGs

After proving the existence of a connected and plane spanning subgraph for graphs in the class RCG, the next task is to analyze if such a connected plane subgraph can be constructed by a local algorithm. Such a local algorithm, the CP (Connected Planarization)-algorithm [3], for the construction of a connected and plane subgraph $G' = (V, F)$ from a given graph $G = (V, E)$ in RCG is presented and investigated in the following. A key property for the correctness of the CP-algorithm for graphs in RCG is that every edge in $E \setminus F$ is intersected by an edge in F. For a graph with this property, alternative paths along convex hulls connecting two vertices can be obtained, following the approach described in Lemma 18. In order to obtain such a path, there has to exist an edge in E between the vertices that are connected by the path. This can be obtained by the property that for an edge uv in F and an edge wx intersecting with uv, at least one of the edges uw or vw and at least one the edges ux and vx exists in E. In the case that only the two edges uw and vx or only the two edges vw and ux exist, the redundancy property is not satisfied for the intersection of uv and wx. So only the two cases remain, where both of the two edges uw and ux or both of the two edges

vw and vx exist. In the CP-algorithm, this property is supposed to hold for all edges $w_i x_i$ intersecting with uv, if uv is an edge in F. The CP-algorithm was initially presented in [3], but the correctness was not proven there.

The CP-algorithm can be executed in a global (Alg. 4) and in a local (Alg. 5) version. The local version of the CP-algorithm, requires only the information in the 2-hop neighborhood of a vertex to decide whether an edge is contained in F or not. In the local approach conflicts among intersecting edges may occur if the corresponding vertices make their decision in parallel. Therefore, the correctness proof in Section 7 is given for the global approach, where such conflicts cannot occur, while the way such conflicts can be handled in the local approach is considered separately.

5.3.1 Algorithm Description

The task of the following CP (Connected Planarization)-algorithm is to construct a connected and plane spanning subgraph $G' = (V, F)$ from a given connected graph $G = (V, E)$. The CP-algorithm starts with a working set W, containing all edges E of G and an empty result set F. As long as the working set is not empty, an edge uv from the working set W is regarded and removed from W. The CP-algorithm decides if the edge uv is added to F by checking the following condition:

Definition 30 (The CP-condition). *The CP-condition is satisfied in a graph $G = (V, E)$ for an edge $uv \in E$ if for all intersecting edges $w_i x_i \in E$ the two redundancy edges uw_i and ux_i or the two redundancy edges vw_i and vx_i exist. (see Fig. 29(**a**)).*

If the CP-condition is satisfied for an edge uv, then the edge is added to F and all edges intersecting with uv are removed from the working set W. Fig. 29 illustrates which edges satisfy the CP-condition and which not.
Since some of the edges can be removed from the working set, without checking the CP-condition for those edges, there are two possibilities why an edge uv is not in F:

(a) An edge wx intersecting with uv was added to F before uv was considered.

(b) The edge uv does not satisfy the CP-condition: It exists an intersecting edge wx such that $(uw \notin E \wedge vw \notin E) \vee (ux \notin E \wedge vx \notin E)$.

In both cases an edge wx exists which prohibits that the edge uv is added to F. The edge uv is therefore deleted from the working set W, without being added to F because of the edge wx. Since one of the conditions (a) or (b) is satisfied, for the intersection of uv and wx, the two redundancy edges wu and wv or the two redundancy edges xu and xv exist. The following definition is used to distinguish, which of the redundancy edges exist for the intersection of uv and wx.

Algorithm 4 Global CP-algorithm for constructing the result graph G'

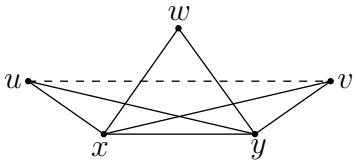
Input: A finite and connected graph $G = (V, E)$ in the class RCG
Output: A connected and plane spanning subgraph $G' = (V, F)$ of G

$F \leftarrow \emptyset$ (* initialize the result *)
$W \leftarrow E$ (* set up the working set *)
while $W \neq \emptyset$ **do**
  choose $uv \in W$
  remove uv from W
  if $\forall w_i x_i \in E$ intersecting with uv $((uw_i \wedge ux_i \in E) \vee (vw_i \wedge vx_i \in E))$
    $\wedge w_i$ not located on $uv \wedge x_i$ not located on uv **then**
    add uv to F
    remove all intersecting edges $w_i x_i$ from W
  end if
end while
$G' \leftarrow (V, F)$

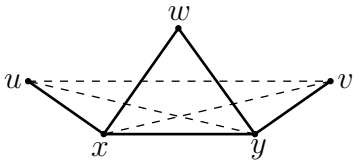
Definition 31 (**Deleting edges**). *The edges uv and wx intersect and one of the conditions (a) or (b) is satisfied for the intersection of uv and wx. If $wu, wv \in E$, then wx* deletes *uv and if $xu, xv \in E$, then xw* deletes *uv.*

The order in which the algorithm chooses the edges from the working set W determines which edges are added to F and which not. Therefore, also the condition for deleting edges depends on the order in which the edges are chosen from the working set. If an edge wx deletes an edge uv due to condition (b), then wx might be itself deleted by another edge. Therefore, deleting edges are not guaranteed to be in F.

However, it can be proven that if an edge uv is not in F because it does not satisfy the CP-condition, then there also exists an edge intersecting with uv that is in F. Therefore, for every edge that is not in F, it can be assumed that it was not added to F because of condition (a). The property, that every edge, which is not in F, is intersected by an edge in F, is proven in Section 7. It has a major role in the correctness proof of the CP-algorithm.



(**a**) The edge uv does not satisfy the CP-condition; all other edges satisfy it.



(**b**) A possible output graph after the execution of the CP-algorithm.

Figure 29: Illustration of the CP-condition and the CP-algorithm.

5.3.2 Local Variant of the CP-Algorithm

The CP-algorithm can be implemented in a distributed version, where only information about the 2-hop-neighborhood of a vertex is required. In order to show that the 2-hop-neighborhood of a vertex is sufficient, the intersecting edges have to be detected first. Under redundancy property intersecting edges can always be detected with 2-hop information from at least one of the two endvertices. Moreover, according to the CP-condition an edge uv can be in F only if for all intersecting edges wx always one of the vertices u or v is connected to w and x. If u cannot detect an intersection of the edge wx with the edge uv with 2-hop information, since neither w nor x are in the neighborhood of u, then u will add this edge to F. Furthermore, if u is not connected to w and x, then the edges vw and vx have to exist. Therefore, uv satisfies the CP condition for the intersection with wx and v will add this edge vu to F as well. Thus, with 2-hop information a distributed CP-algorithm can be implemented as depicted by Alg. 5.

Algorithm 5 Distributed version of the algorithm from the view of a vertex u

Input: A finite and connected graph $G = (V, E)$ in the class RCG
Output: A connected and plane spanning subgraph $G' = (V, F)$ of G

```
1: F ← ∅ (* initialize the result for the local view of u *)
2: W ← Edges connecting u with its two hop neighborhood (* set up the working set *)
3: for v ∈ N(u) do
4:     c(uv) = true (condition for adding the edge uv)
5:     for w ∈ N(u) do
6:         for x ∈ N(w) do
7:             if wx intersects uv and (vx ∉ E and ux ∉ E)
8:                 or w is located on uv or x is located on uv then
9:                 c(uv) = false
10:            end if
11:        end for
12:    end for
13:    if c(uv) = true then
14:        add uv to F
15:        inform w and x of all edges wx ∈ W intersecting uv to remove wx
16:    else
17:        inform all neighbors to remove uv
18:    end if
19: end for
20: G'(u) ← (N(u), F)
```

However, in order to avoid conflicting distributed decisions about edge removals, the algorithm requires an underlying distributed serialization of edge decisions. Such serialization can be based on a total ordering defined on the edges. When the edges are regarded in the same order, then the edge set F of the global and the local algorithm is equal, since the condition for adding an edge in F is the same in both algorithms.

6 Modelling Geometric Graphs

In order to find a suitable automated reasoning approach for the correctness proof of the CP-algorithm it is first regarded how graphs and graph algorithms are studied so far, when they methods from automated reasoning are used. Such approaches are often based on monadic second-order logic or higher-order logic (HOL). Formalizations of graph properties in Isabelle/HOL can be found in [56, 43]. Approaches using monadic second-order logic were studied by Courcelle in many of his publications, for example in [18, 19, 20, 21]. Many of the studies of graphs and their algorithms in automated reasoning have in common that they regard graphs with bounded tree-width or bounded clique-width. These are properties, which do not necessarily hold for classes of graphs studied in the context of the CP-algorithm. Furthermore, research that combines reasoning about graphs and reasoning about geometric properties of graphs is not known so far. Therefore, an approach that is orthogonal to the existing approaches for automated reasoning on graphs and graph algorithms is proposed.

For describing and proving the correctness of the CP-algorithm geometric graphs have to be studied. In order to avoid higher-order logic and higher-order logic systems it is regarded first, which aspects of geometry and geometric graphs can be described in first-order-logic. For the type of considered graphs (geometric graphs with vertices in the Euclidean plane) and for the considered verification problems on those graphs the use of second-order logic can be largely avoided. The main challenge here is handling connectivity, a property known to be not first-order definable. In order to avoid the use of HOL, uniquely determined paths along the convex hull of determined sets of points (usually vertices) are considered. Those paths are represented using linked data structures. Considering those paths is sufficient to prove the connectivity of a graph.
Parts of the results presented in this section were already published in [14]

6.1 Reasoning in Geometry and Graph Theory

When modeling and reasoning about geometric graphs, geometric aspects have to be considered in addition to the encoding of the topological aspects of graphs, i.e. vertices and edges. Reasoning about the graphs themselves has to be combined with reasoning about geometric properties. There exist different approaches for reasoning in geometry. One approach is using coordinates and ultimately analytic geometry, another approach is an axiomatic one. For the approaches using automated reasoning in *analytic geometry*, methods based on Gröbner bases (cf. e.g. [22]) or approaches based on quantifier elimination in real closed fields (cf. e.g. [72, 26]) exist. The approaches of using *axioms for plane geometry* have been proposed by systems of Tarski [74] or Hilbert [36]. For theorem proving in geometry such axiom systems were used for instance by Beeson and Wos in [4].

The CP-algorithm is described and proven to be correct, by using an axiomatic approach. However, only a small part of plane geometry has to be expressed for this task. Compared to

classical axiom systems for plane geometry, such as those proposed by Tarski [74] and Hilbert [36], only axioms of order are used, but axioms for incidence, congruence, parallelity and continuity are not included. The axioms of order are sufficient, because the goal is to describe only the part of plane Euclidean geometry, which is necessary to describe geometrically bounded graph drawings in the Euclidean plane. Due to technical considerations (which are explained in Section 6.1), the proposed axioms are slightly different from the axioms of Hilbert [36], but all necessary conditions to prove the correctness of the algorithm are satisfied.

An advantage of the proposed axiomatic approach of plane geometry is that the axiom system is simple and only information about the relative positions of vertices is required. A relative position refers to an area defined by other vertices, whereby the exact position within this area is not important. Therefore, concrete coordinates, lengths and angles between the vertices do not have to be considered. So the proposed axiom system is stable for small changes of the position of the vertices, where the relative positions of the vertices, (i.e. whether edges intersect or not and whether a vertex is located inside a triangle formed by other vertices or not), do not change. In case that the relative positions do not change, also the predicates do not change. The simple axiom system makes it easier for a prover to decide whether a condition for a specific set of predicates holds or not.

The proposed axiom system is used for the correctness proof of the CP-algorithm, which is based on a manually created a series of implications. For those implications, the correctness is verified by automatically proving that the negation of an implication is unsatisfiable. Putting those implications together in a general manually derived proof scheme proves the correctness of the considered proof tasks. If for a proof task unsatisfiability cannot be proven, the possibilities of generating models with Z3 are used. Those models can be used, to show that the considered axioms cannot be relaxed further or proof tasks cannot be simplified.

6.2 An Axiomatic system for RCGs and the CP-algorithm

For automatically verifying the correctness proof of the CP-algorithm, an efficient way of representing geometric graphs and their properties, in particular the redundancy and coexistence property has to be found. For expressing the redundancy property, the property that two edges geometrically intersect when they are regarded as segments, has to be expressed. For expressing the coexistence property the property that a point is in the interior of a triangle has to be expressed. An axiomatic approach can be used to distinguish if two lines intersect or if a point is located inside the triangle formed by three other points. Since only the positions of the vertices u and v forming an edge uv are relevant, but not of points located on this edge, all proof tasks can be adapted for graphs, where the straight line segments are replaced by arbitrary curves, as long as all intersection resp. inside relations still hold.

In what follows an axiomatization is given and it is shown that this axiomatzation is sufficient for automatically verifying the proof steps in the correctness proof of the CP-

algorithm. In the proposed approach first axioms describing geometric aspects are defined, then axioms describing graph properties are added. Next notions and constructions needed for describing the CP-algorithm are defined and finally notions and constructions for the correctness proof of the CP-algorithm are added.

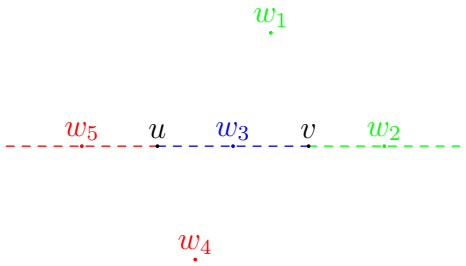
6.2.1 Geometric Axioms

The geometric axioms are structured in a hierarchical way, starting with an axiomatization of the properties of a predicate left of arity 3. Using this predicate additional notions are axiomatized (intersection, inside).

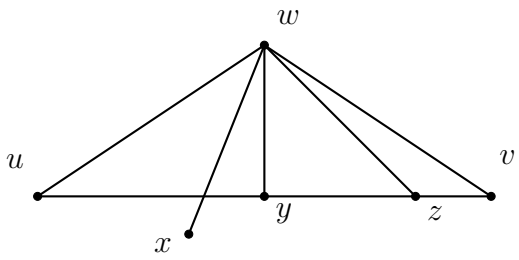
The idea of the definition of a predicate left is to divide a plane into two half planes, a left and a right half plane, by a line passing through two points a and b. For every point c, a predicate $\mathsf{left}(a, b, c)$ has to be defined in such a way, that $\mathsf{left}(a, b, c)$ is true iff c is in the left half plane defined by the line passing through a and b. Analogously the right half plane has to be defined. In order to simplify the definition instead of a predicate $\mathsf{right}(a, b, c)$ the right half plane of the line passing through a and b is defined as the left half plane of the line passing through b and a.

So the predicate $\mathsf{left}(a, b, c)$ can also be regarded as a relation of the angle $\angle bac$ between the segments ab and ac. Then $\mathsf{left}(a, b, c)$ is true if $\angle bac \in (0, \pi)$. But it has to be investigated if $\mathsf{left}(a, b, c)$ should also be true if a, b, c are colinear, i.e. $\angle bac = 0$, $\angle bac = \pi$, or $a = b$. Those cases have to regarded in detail, in order to obtain a correct definition of the intersection of two segments and the interior of a triangle.

Intersection of segments (see Fig. 30(**b**)) According to the definition for plane graphs two segments uv and wx intersect iff they have a point p in common, which is not an end point of both of the segments.



(**a**) w_1, w_2 and w_3 are left of uv, while w_3, w_4 and w_5 are left of vu.



(**b**) Both, y and z are located inside the triangle Δuvw, furthermore wx and wy and wz are intersecting uv

Figure 30: Illustration of the predicates left, intersection and inside

In the following, all possible cases of colinear and not colinear vertices are considered, in order to find a suitable definition of the predicate left, which can be used to express the condition for an intersection correctly.

Case 1: First the case that there are no collinear points among the four points u, v, w, x is considered. If the segments uv and wx intersect, they have a unique point p in common. Furthermore, one of the points w or x is located left of the line passing through u and v and one of the points w or x is located left of the line passing through v and u. Analogously one of the points u or v is located left of the line passing through w and x and one of the points u or v is located left of the line passing through x and w.

Therefore, the predicate left can be used to define intersections between segments, where the considered points are pairwise not collinear. A suggested definition for an intersection is therefore, that uv and wx are intersecting iff one of the points u and v is left of the line passing through w and x and one left of the line passing through x and w, as well as one of the points w and x is left of the line passing through u and v and one left of the line passing through v and u. It is now regarded in the following cases, whether the predicate left can be defined in a way, such that the suggested definition of an intersection still holds for the limit cases of equal or collinear points.

Case 2: Now the case of three not collinear points u, v and w is considered. The segments uv and uw are not intersecting in this case, since the only point the two segments have in common is u. Let the labeling of the points be chosen, such that w is left of the line passing through u and v and v is left of the line passing through w and u. If the predicate $\mathsf{left}(a, b, c)$ would be true for all points c located on the line passing through a and b, then the segments uv and uw would intersect according to the suggested definition, because the predicates $\mathsf{left}(u, v, w)$, $\mathsf{left}(v, u, u)$, $\mathsf{left}(u, w, u)$ and $\mathsf{left}(w, u, v)$ would be all true. Therefore, the predicate $\mathsf{left}(a, b, c)$ cannot be true for all points c located on the line passing through a and b.

Case 3: Next the case of three colinear points u, v and x, where the labeling is chosen such that x is located between u and v, is considered. The two segments uv and xv have the point x in common, which is only an end point of the segment xv, but not an end point of the segment uv. Therefore, the segments uv and xv are intersecting. The suggested condition for the intersection is satisfied if the predicates $\mathsf{left}(u, v, x)$, $\mathsf{left}(v, u, x)$, $\mathsf{left}(v, x, u)$ and $\mathsf{left}(x, v, v)$ are true. This can be obtained if the predicate $\mathsf{left}(a, b, c)$ is true for all c located on the ray starting in a and passing through b resp. when regarding the angle, if $\angle bac = 0$.

This definition of the predicate left is obviously consistent with the suggested definition of an intersection for Case 1, but for Case 2 this needs to be examined more closely. For the situation of Case 2, according to the suggested definition of the predicate left, the predicate is true for $\mathsf{left}(u, v, w)$, $\mathsf{left}(v, u, u)$ and $\mathsf{left}(w, u, v)$. If the predicate $\mathsf{left}(u, w, u)$ would be also true, then the segments uv and uw would intersect. Therefore, the suggested definition of the predicate left is consistent with the suggested definition of an intersection for Case2, if the restriction that $\mathsf{left}(a, b, a)$ is false is added.

Next, the suggested definition of the predicate left is regarded for the remaining cases.

For Case 3 it remains to consider the segments ux and vx, which are not intersecting. This condition is satisfied, because $\mathsf{left}(x, u, v)$ and $\mathsf{left}(x, u, x)$ are both false.

Case 4: Next the case that three points u, v and y are colinear, with y located between u and v, and a point w, which is not colinear with u, v and y is considered. The segments uv and wy are intersecting. This condition is satisfied, since $\mathsf{left}(u, v, y)$ and $\mathsf{left}(v, u, y)$ are true and furthermore both predicates $\mathsf{left}(w, y, u)$ and $\mathsf{left}(y, w, v)$ or both predicates $\mathsf{left}(w, y, v)$ and $\mathsf{left}(y, w, u)$ are true. On the other hand uw and vy are not intersecting. Also this condition is satisfied, since both predicates $\mathsf{left}(u, w, v)$ and $\mathsf{left}(u, w, y)$ are false. Analogously the segments vw and uy can be regarded.

Case 5: Finally it remains the case of four colinear points u, v, y and z, where the labeling is choosen such that, y is located between u and v and furthermore z is located between y and v. Then the segments uv and yz are intersecting. This condition is satisfied, because the predicates $\mathsf{left}(u, v, y)$, $\mathsf{left}(v, u, y)$, $\mathsf{left}(y, z, v)$ and $\mathsf{left}(z, y, u)$ are true. Furthermore, the segments uz and yv are intersecting. Also this condition is satisfied, since $\mathsf{left}(u, z, y)$, $\mathsf{left}(z, u, y)$, $\mathsf{left}(v, y, u)$ and $\mathsf{left}(y, v, z)$ are true. However, the segments uy and vz are not intersecting. Also this condition is satisfied, because the predicates $\mathsf{left}(z, v, u)$ and $\mathsf{left}(z, v, y)$ are both false.

After regarding all possible cases it turns out that the suggested definition of the predicate left, where $\mathsf{left}(a, b, c)$ is true iff c is on the left side of the line passing through a and b or on the ray starting in a and passing through b, where $c = a$ is excluded, can be used for the definition of intersections.

Points inside a triangle: Now the second important condition defined by the predicate left is regarded. This is the condition if a point x is located inside a triangle formed by u, v and w. According to the definition of the interior of a triangle, the border segments uv, vw and wu are part of the interior, but the points u, v and w are not part of the interior. A possible definition of a predicate inside is therefore that $\mathsf{inside}(u, v, w, x)$ is true iff $\mathsf{left}(u, v, x)$ and $\mathsf{left}(v, w, x)$ and $\mathsf{left}(w, u, x)$ are true. If the predicate $\mathsf{inside}(u, v, w, x)$ is true, then x is located inside the triangle Δuvw. With this definition a point x can only be located inside a triangle Δuvw if the points u, v, w and x are pairwise distinct, because otherwise one of the predicates $\mathsf{left}(u, v, x)$, $\mathsf{left}(v, w, x)$ or $\mathsf{left}(w, u, x)$ is false.

Next, a point x in the interior of the triangle formed by the points u, v and w is regarded. The point x is located inside the triangle Δuvw if u, v and w are not collinear and w is located left of the line passing through u and v. On the other hand x is located inside the triangle Δvuw if w is located left of the line passing through v and u. For the case of three collinear points u, v and y, where the labeling is chosen such that y is located between u and v, the point z is located inside the triangle Δuvy if z is located between u and y. On the other hand z is located inside the triangle Δvuy if z is located between y and v (see Fig. 30(**b**)) .

After regarding all possible cases, which have to be considered to decide if a point x is located inside a triangle Δuvw, the definition of the predicate left can also be used to define the a predicate inside.

In the following, an axiom system is defined that is consistent, with the suggested definitions of the predicates left, intersection and inside.

Axioms for left. The intended meaning for $\mathsf{left}(u, v, w)$ is *"w is on the left side of the (oriented) line through u and v or on the ray starting in u and passing through v, but $u \neq w$"* see Fig. 30(**a**). This includes the case that $v = w$ (if $u \neq v$). When regarding angles in the Euclidean plane $\mathsf{left}(u, v, w)$ corresponds to $\angle vuw \in [0, \pi)$. Note that $\mathsf{left}(u, v, w)$ is false for $u = v$ and $u = w$. If $u \neq v$ then for every point w at least one of of the predicates $\mathsf{left}(u, v, w)$ or $\mathsf{left}(v, u, w)$ is true. If both predicates $\mathsf{left}(u, v, w)$ and $\mathsf{left}(v, u, w)$ are true then w is located strictly between u and v.
The predicate left has to satisfy the following conditions:

- For $u \neq w$ and $v \neq w$, the point w is located between u and v iff u is not located left of the line passing through w and v as well as v is not located left of the line passing through w and u (A1-2).
- If w is located between u and v, then for a fourth point x, v is located left of a line passing through u and x iff w is located left of a line passing through u and x (A3-4).
- For three not collinear points u, v and w, a point x is located on the same side of a line passing through u and v as w, on the same side of a line passing through v and w as u or on the same side of a line passing through w and u as v (A5).
- If the border points of a triangle are left of a line passing through x and y, then also a point z located inside this triangle is located left of a line passing through x and y (A6).

These conditions are all independent of each other and therefore necessary in axioms defining the predicate left. Following those conditions, the axioms defining the predicate left are given below (the variables u, v, w, x, y, z are universally quantified), see also Fig. 30(**a**).

A1 : $(\mathsf{left}(u, v, w) \wedge \mathsf{left}(v, u, w) \rightarrow \neg\mathsf{left}(w, u, v))$
A2 : $(u \neq w \wedge v \neq w \wedge \neg\mathsf{left}(w, u, v) \wedge \neg\mathsf{left}(w, v, u) \rightarrow \mathsf{left}(u, v, w))$
A3 : $(\mathsf{left}(u, v, w) \wedge \mathsf{left}(v, u, w) \wedge \mathsf{left}(u, x, w) \rightarrow \mathsf{left}(u, x, v))$
A4 : $(\mathsf{left}(u, v, w) \wedge \mathsf{left}(v, u, w) \wedge \mathsf{left}(u, x, v) \rightarrow \mathsf{left}(u, x, w))$
A5 : $(\mathsf{left}(u, v, w) \wedge \mathsf{left}(v, w, u) \wedge \mathsf{left}(w, u, v) \rightarrow \mathsf{left}(u, v, x) \vee \mathsf{left}(v, w, x) \vee \mathsf{left}(w, u, x))$
A6 : $(\mathsf{left}(u, v, z) \wedge \mathsf{left}(v, w, z) \wedge \mathsf{left}(w, u, z) \wedge$
$\mathsf{left}(x, y, u) \wedge \mathsf{left}(x, y, v) \wedge \mathsf{left}(x, y, w) \rightarrow \mathsf{left}(x, y, z))$

From **A1, A2, A3, A4, A5, A6** further conditions for the predicate left can be derived, that are supposed to be satisfied in Euclidean geometry according to the intended meaning

of the predicate left. It is therefore not required to add those conditions as axioms. Those derived conditions can be found in Appendix 9.3.1 and include the conditions of the limit cases for $\mathsf{left}(u, v, w)$, when $u = v$, $u = w$ or $v = w$. Therefore, from the axioms A1-6 it already follows that for all u and v the predicates $\mathsf{left}(u, u, v)$ and $\mathsf{left}(u, v, u)$ are always false and the predicate $\mathsf{left}(u, v, v)$ is true if $u \neq v$. Axiom $A5$ can be derived from the other axioms if there exist at least 5 different points. Furthermore, the axioms from Hilbert in the group of order can be derived from the axioms A1-6. So choosing the axioms A1-6 is also sufficient to describe the ordering of points in the Euclidean plane according to Hilbert's axiom system. Since proving the axioms from Hilbert requires a definition of two additional predicates, intersection and inside of arity 4. Those two predicates are defined first based on the predicate left.

Axioms for intersection: The intended meaning for $\mathsf{intersection}(u, v, w, x)$ is *"the segments uv and wx intersect"*. This is the case iff one of the points w or x is located left of uv and one left of vu and one of the points u or v is located left of wx and one left of xw.

This definition is consistent with the definition, that two segments intersect if they have a point in common, which is not an end vertex of both segments. Moreover, a segment cannot intersect with itself, therefore $\mathsf{intersection}(u, v, u, v)$ and $\mathsf{intersection}(u, v, v, u)$ are false. The axioms defining the intersection predicate (names starting with **I**) are given below (the variables u, v, w, x are universally quantified).[3]

I1-2 : $\mathsf{intersection}(u, v, w, x) \rightarrow (u \neq w \vee v \neq x) \wedge (u \neq x \vee v \neq w)$

I3 : $\mathsf{intersection}(u, v, w, x) \rightarrow \mathsf{left}(u, v, w) \vee \mathsf{left}(u, v, x)$

I4 : $\mathsf{intersection}(u, v, w, x) \rightarrow \mathsf{left}(v, u, w) \vee \mathsf{left}(v, u, x)$

I5 : $\mathsf{intersection}(u, v, w, x) \rightarrow \mathsf{left}(w, x, u) \vee \mathsf{left}(w, x, v)$

I6 : $\mathsf{intersection}(u, v, w, x) \rightarrow \mathsf{left}(x, w, u) \vee \mathsf{left}(x, w, v)$

I11-12 : $\mathsf{left}(u, v, w) \wedge \mathsf{left}(v, u, x) \wedge \mathsf{left}(w, x, u) \wedge \mathsf{left}(x, w, v) \rightarrow$
$\mathsf{intersection}(u, v, w, x) \vee (u = x \wedge v = w)$

I13 : $\mathsf{left}(u, v, w) \wedge \mathsf{left}(v, u, x) \wedge \mathsf{left}(w, x, v) \wedge \mathsf{left}(x, w, u) \rightarrow \mathsf{intersection}(u, v, w, x)$

I14 : $\mathsf{left}(u, v, x) \wedge \mathsf{left}(v, u, w) \wedge \mathsf{left}(w, x, u) \wedge \mathsf{left}(x, w, v) \rightarrow \mathsf{intersection}(u, v, w, x)$

I15-16 : $\mathsf{left}(u, v, x) \wedge \mathsf{left}(v, u, w) \wedge \mathsf{left}(w, x, v) \wedge \mathsf{left}(x, w, u) \rightarrow$
$\mathsf{intersection}(u, v, w, x) \vee (u = w \wedge v = x)$

Axioms for inside: The intended meaning for $\mathsf{inside}(u, v, w, x)$ is *"the point x is located inside or on the boundary the triangle Δuvw, but x is not equal to u,v or w"*. Therefore, inside can be defined by the following axioms (the variables u, v, w, x are universally quantified):

T1-4 : $(\mathsf{inside}(u, v, w, x) \leftrightarrow \mathsf{left}(u, v, x) \wedge \mathsf{left}(v, w, x) \wedge \mathsf{left}(w, u, x))$

The considered triangles are oriented, so the order in which the vertices are written is important: A point x is located inside the triangle Δuvw iff $\mathsf{left}(u, v, x)$, $\mathsf{left}(v, w, x)$ and

[3]Since the clause form of some of the axioms contain more than one formula, and for doing the tests axioms are used in clause form. In such cases a range of numbers is used for the axioms.

$\mathsf{left}(w,u,x)$ are all true; x is located inside the triangle Δuwv iff $\mathsf{left}(u,w,x)$, $\mathsf{left}(w,v,x)$ and $\mathsf{left}(v,u,x)$ are all true. The triangle Δuvw is equal to the triangles Δvwu and Δwuv. It can be proven:

- The inside predicate is transitive, i.e. if y is located inside the triangle Δuvx and x is located inside the triangle Δuvw, then y is also located inside the triangle Δuvw (**T11-T13**).
- If x and y are located inside a triangle Δuvw, where $x \neq y$, then y is also located in at least one of the triangles Δuvx, Δvwx or Δwux (**T14**).

In what follows it is refered to the union of the axioms for the predcicates left, intersection and inside by AxGeom. A list of all the considered axioms can be found in Appendix 9.3.1–9.3.3.

Link to other axiom systems for plane geometry. Tarski's [74] and Hilbert's [36] axiom systems for plane geometry use both a predicate "between". Tarski's betweenness notion is non-strict: a point c is between a and b iff c is located on the closed segment $[ab]$, while Hilbert's notion is strict: c is between a and b iff c is located on the open segment (ab). The relation "c is between a and b" can be expressed in the proposed axiomatization by "$\mathsf{left}(a,b,c) \wedge \mathsf{left}(b,a,c)$" and is strict, so Hilbert's axiomatization is closer to the proposed one than Tarski's. Since only line segments are considered, the axioms of Hilbert's group 1 (incidence), 3 (congruence), 4 (parallels) and 5(continuity) (cf. [36]) are not important for the proposed axiomatization. The related axioms can be found in group 2 of Hilbert's axioms, which contains the following axioms:

- **Axiom 2.1:** If a point B is between points A and C, B is also between C and A, and there exists a line containing the distinct points A, B, C.
- **Axiom 2.2:** If A and C are two points, then there exists at least one point B on the line AC such that C is between A and B.
- **Axiom 2.3:** Of any three points situated on a line, there is no more than one which is between the other two.
- **Axiom 2.4: (Pasch's axiom):** Let A, B, C be three points not being on the same line and let a be a line in the plane ABC and not passing through any of the points A, B, C. Then, if the line a passes through a point of the segment AB, it will also pass through either a point of the segment BC or a point of the segment AC.

Hilbert's axioms 2.1, 2.3 and 2.4 can be derived from the proposed axioms. Since only a finite set of points (related to the vertices of the considered graphs or to intersections of the edges in the considered graphs) is considered, Hilbert's axiom 2.2 (containing a $\forall\exists$-sentence) was not needed for the correctness proof, so no equivalent was included. The axiom 2.1 can be trivialy

derived from the proposed axioms, since $\mathsf{left}(a,b,c) \wedge \mathsf{left}(b,a,c) \leftrightarrow \mathsf{left}(b,a,c) \wedge \mathsf{left}(a,b,c)$. The axiom 2.3 follows directly from **A1** and **A2**. The implication of Pasch's axiom from **A1-6** is proven in Section 6.3.3. Pasch's axiom could also be added in the axiom set and replace the axioms **A4** and **A6**, because they can be derived by the remaining axioms of AxGeom and Pasch's axiom. However, the axioms are used in the proposed way, because the axioms defining the predicate left can be introduced independently first and afterwards the axioms defing the predicates intersection and inside can be formulated based on the axioms defining the predicate left.

6.2.2 Axiomatizing Graph Properties

In the next step conditions for vertices and edges are defined. A unary predicate V is used for representing the vertices[4] and a binary predicate E for representing the edges. Axioms for the edge predicate (denoted by **E**) are given below (the variables u, v are universally quantified):

E1:	$\neg\mathsf{E}(u,u)$	(self-loops are not allowed)
E2:	$\mathsf{E}(u,v) \rightarrow \mathsf{E}(v,u)$	(E is symmetric)
E3:	$\mathsf{E}(u,v) \rightarrow \mathsf{V}(u) \wedge \mathsf{V}(v)$	(edges exist only between vertices)

Based on the definitions of the predicates intersection, inside and V and E the conditions for the two properties redundancy and coexistence are defined. The redundancy property expresses the condition that if two edges uv and wx intersect, then one of the vertices u, v, w or x is connected to the other 3 ones. This condition is equivalent to the condition that one of the edges uw or vx and one of the edges ux and vx has to exist. By the symmetry of the edges only one condition is required. The redundancy property is therefore expressed with the following axiom (the variables u, v, w, x are universally quantified):

R1 : $\mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{E}(u,w) \vee \mathsf{E}(v,x)$

The coexistence property expresses the condition that if x is a vertex, $\mathsf{inside}(u,v,w,x)$ is true and the edges uv, vw and wu exist, then also the edges ux, vx and wx exist. By the rotation of the triangle it is sufficient that only one of the edges has to exist. Axiom **C1** (the variables u, v, w, x are universally quantified) expresses this:

C1 : $\mathsf{inside}(u,v,w,x) \wedge \mathsf{E}(u,v) \wedge \mathsf{E}(v,w) \wedge \mathsf{E}(w,u) \wedge \mathsf{V}(x) \rightarrow \mathsf{E}(u,x)$

6.2.3 Axioms for Notions Used in the CP-algorithm

For graphs satisfying the redundancy and coexistence property the CP-algorithm constructs a connected and plane subgraph $G' = (V, F)$. For the edge set F a binary predicate F is used. The properties of this predicate are described by a set of axioms **F1**–**F6** (in these formulae the variables u, v, w, x are universally quantified). Axioms **F1**–**F3** express that the edges in

[4]The predicate V for the vertices is listed here to distinguish between the axioms for geometry, which hold for arbitrary points, and properties for edges between vertices in a graph.

F are a subset of E and F is symmetric and intersection-free. Axiom **F4** expresses that no vertex can be located on an edge in F.

F1: $\mathsf{F}(u,v) \rightarrow \mathsf{F}(v,u)$
F2: $\mathsf{F}(u,v) \rightarrow \mathsf{E}(u,v)$
F3: $\mathsf{F}(u,v) \wedge \mathsf{F}(w,x) \rightarrow \neg\mathsf{intersection}(u,v,w,x)$
In addition, no vertex can be located on an edge in F:
F4: $\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{V}(w) \rightarrow \neg\mathsf{F}(u,v)$

Furthermore, the edges in F have to satisfy the CP-condition, i.e. for each edge wx intersecting with an edge uv in F, both edges uw and ux or both edges vw and vx have to exist. This is expressed (after removing the implications already entailed by **R1** and **F2**) by the following axioms:

F5: $\mathsf{F}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{E}(u,w) \vee \mathsf{E}(v,w)$ (CP-condition)
F6: $\mathsf{F}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{E}(u,x) \vee \mathsf{E}(v,x)$ (CP-condition)
(only one of the conditions **F5** or **F6** is needed due to symmetry)

6.2.4 Axioms for Notions Used in the Correctness Proof

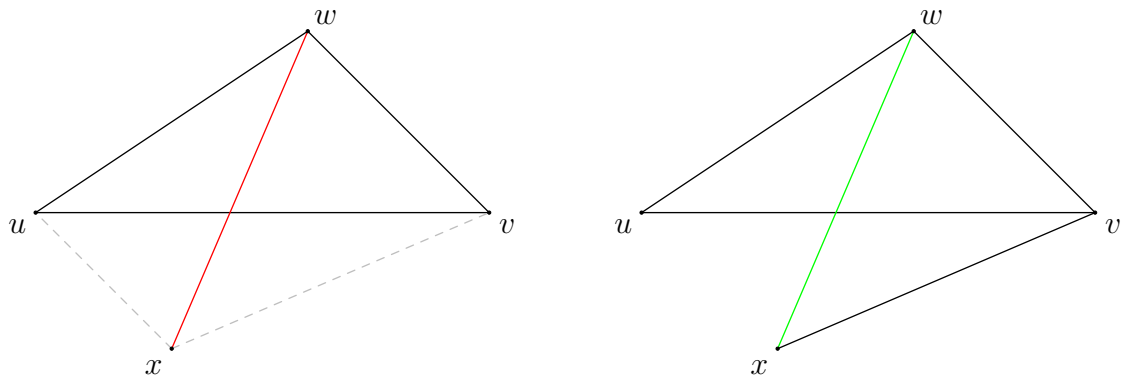
For the correctness proof a further predicate **deleting** of arity 4 is defined, and notions for a path along the convex hull of points in a certain triangle are formalized. The used axiomatizations for this are described below.

Axioms for deleting: The intended meaning for $\mathsf{deleting}(u,v,w,x)$ is *"uv and wx are intersecting edges, where wx prohibits that uv is in F, while the redundancy edges wu and wv exist."*.

To determine the existence of redundancy edges, the order of the vertices w and x is relevant. Therefore, if $\mathsf{deleting}(u,v,w,x)$ is true then the two redundancy edges wu and wv exist, while if $\mathsf{deleting}(u,v,x,w)$ is true then the two redundancy edges xu and xv exist. The edge wx can prohibit that the edge uv is in F if wx is in F or if only the redundancy edges wu and wv, but neither xu nor xv exist for the intersection of uv and wx. Thus, $\mathsf{deleting}(u,v,w,x) \leftrightarrow \mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \wedge \mathsf{E}(u,w) \wedge \mathsf{E}(v,w) \wedge \big((\neg\mathsf{E}(u,x) \wedge \neg\mathsf{E}(v,x)) \vee \mathsf{F}(w,x)\big)$. The axioms defining the deleting predicate are denoted by **D** and represent the condition above (the variables u,v,w,x are universally quantified).

D1-3: $\mathsf{deleting}(u,v,w,x) \rightarrow \mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x)$
D4-5: $\mathsf{deleting}(u,v,w,x) \rightarrow \mathsf{E}(u,w) \wedge \mathsf{E}(v,w)$
D11: $\mathsf{E}(u,x) \wedge \mathsf{deleting}(u,v,w,x) \rightarrow \mathsf{F}(w,x)$
D12: $\mathsf{E}(v,x) \wedge \mathsf{deleting}(u,v,w,x) \rightarrow \mathsf{F}(w,x)$
D13: $\mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{deleting}(u,v,w,x) \vee \mathsf{E}(u,x) \vee \mathsf{E}(v,x)$
D14: $\mathsf{E}(u,v) \wedge \mathsf{F}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \wedge \mathsf{E}(u,w) \wedge \mathsf{E}(v,w) \rightarrow \mathsf{deleting}(u,v,w,x)$

In what follows, it is refereed to the axioms describing properties of E and F, and of the predicate deleting (axioms **E1**–**E3**, **F1**–**F6**, and **D1**–**D14**) by AxGraphs.



(**a**) The edge wx is deleting the edge uv, since $ux \notin E$ and $vx \notin E$

(**b**) The edge wx is in F and deleting the edge uv

Figure 31: Two possibilities, where $\mathsf{deleting}(u, v, w, x)$ is true

In the correctness proof usually only an induced subgraph for a subset of the vertices of G is regarded. Therefore, an edge uv is assumed to be in F only if this follows already from the predicates for the considered subgraph. If this is not the case, the edge uv is assumed to be not in F, because it might be deleted by an edge wx in G, where at least one of the vertices w or x is not part of the currently regarded subgraph. In the figures illustrating graph examples, deleting edges are represented in red, if they are not necessarily in F, while they are represented in green if they are guaranteed to be in F (see Fig. 31). Furthermore, segments between vertices, where explicitly no edge exists are represented by a dashed line. In some of the examples, edges might be omitted in the figure due to better visibility. Therefore, sometimes the redundancy or coexistence property might not be satisfied for graphs illustrated in a figure, but the corresponding graph considered in the proof is indeed a RCG.

Path along the convex hull of a set of points. In the following, for a pair of intersecting edges uv and wx with intersection point q, the triangle $\Delta u_1 q w_1$ is considered. The labeling of the vertices is chosen such that the vertex w_1 is located left of $u_1 v_1$, the vertex x_1 is located left of $v_1 u_1$, the vertex u_1 is located left of $x_1 w_1$ and the vertex v_1 is located left of $w_1 x_1$. For representing a path from u_1 to w_1 along the convex hull of all points located inside of the triangle $\Delta u_1 q w_1$, a list structure with a function symbol next is used. For every point i, $\mathsf{next}(i)$ is the next vertex on this path if i is a vertex on the convex hull that has a successor, or nil otherwise. Axioms **Y0**–**Y4** express the properties of the path along the convex hull. The end of this path is defined as w_1 by setting $\mathsf{next}(w_1) = \mathsf{nil}$ **(Y0)**. For all vertices y, different from i and $\mathsf{next}(i)$, the vertex y is left of the line passing through i and $\mathsf{next}(i)$ or i is between y and $\mathsf{next}(i)$ **(Y1-2)**. Furthermore, for a vertex y, different from i and $\mathsf{next}(i)$, that is located left of the line passing through $\mathsf{next}(i)$ and i, the vertex i is between y and $\mathsf{next}(i)$ **(Y3-4)**.

The axioms **Y0**–**Y4** are defined in the following way, where the universal quantifiers are explicitly written.

$\mathbf{Y0}:\ \mathsf{next}(w_1) = \mathsf{nil}$

$\mathbf{Y1}:\ \forall y, i\ (\mathsf{V}(y) \wedge \mathsf{left}(u_1, v_1, y) \wedge \mathsf{left}(x_1, w_1, y) \rightarrow i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee$
$y = i \vee y = \mathsf{next}(i) \vee \mathsf{left}(i, \mathsf{next}(i), y) \vee \mathsf{left}(y, \mathsf{next}(i), i))$

$\mathbf{Y2}:\ \forall y, i\ (\mathsf{V}(y) \wedge \mathsf{left}(u_1, v_1, y) \wedge \mathsf{left}(x_1, w_1, y) \rightarrow i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee$
$y = i \vee y = \mathsf{next}(i) \vee \mathsf{left}(i, \mathsf{next}(i), y) \vee \mathsf{left}(\mathsf{next}(i), y, i))$

$\mathbf{Y3}:\ \forall y, i\ (\mathsf{V}(y) \wedge \mathsf{left}(u_1, v_1, y) \wedge \mathsf{left}(x_1, w_1, y) \wedge \mathsf{left}(\mathsf{next}(i), i, y) \rightarrow$
$i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee y = i \vee y = \mathsf{next}(i) \vee \mathsf{left}(y, \mathsf{next}(i), i))$

$\mathbf{Y4}:\ \forall y, i\ (\mathsf{V}(y) \wedge \mathsf{left}(u_1, v_1, y) \wedge \mathsf{left}(x_1, w_1, y) \wedge \mathsf{left}(\mathsf{next}(i), i, y) \rightarrow$
$i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee y = i \vee y = \mathsf{next}(i) \vee \mathsf{left}(\mathsf{next}(i), y, i))$

Axioms **Y11**–**Y13** express the condition that all the vertices on the path along the convex hull are located inside the triangle $\Delta u_1 q w_1$ and that there exists an edge between the vertices of the convex hull.

Y11: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(u_1, v_1, w_1, \mathsf{next}(i)) \vee \mathsf{next}(i) = w_1)$
Y12: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(u_1, x_1, w_1, i) \vee i = u_1)$
Y13: $\forall i\ (\mathsf{E}(i, \mathsf{next}(i)) \vee i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil})$

Axioms **Y21**–**Y24** express conditions for the position of i with respect to u_1, v_1, x_1 and $\mathsf{next}(i)$ as well as conditions for the position of $\mathsf{next}(i)$ with respect to u_1, v_1, x_1 and i. Any vertex i on the path along the convex hull is located inside the triangles $\Delta u_1 v_1 \mathsf{next}(i)$ and $\Delta u_1 x_1 \mathsf{next}(i)$ or equal to u_1 **(Y21-22)** (see Fig. 32(**a**)). Furthermore, the vertex $\mathsf{next}(i)$ is located inside the triangles $\Delta v_1 w_1 i$ and $\Delta x_1 w_1 i$ or equal to w_1 **(Y23-24)** (see Fig. 32(**a**)).

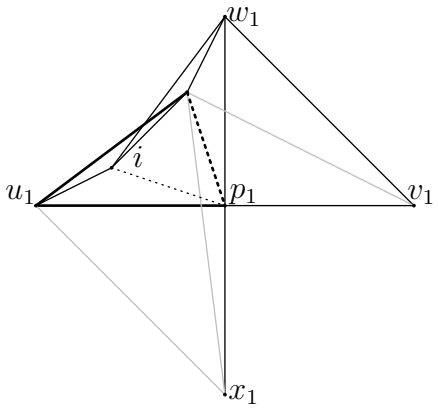
Y21: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(u_1, v_1, \mathsf{next}(i), i) \vee i = u_1)$
Y22: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(u_1, x_1, \mathsf{next}(i), i) \vee i = u_1)$
Y23: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(v_1, w_1, i, \mathsf{next}(i)) \vee \mathsf{next}(i) = w_1)$
Y24: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(x_1, w_1, i, \mathsf{next}(i)) \vee \mathsf{next}(i) = w_1)$

6.3 Analyzing the Axioms for the Verification

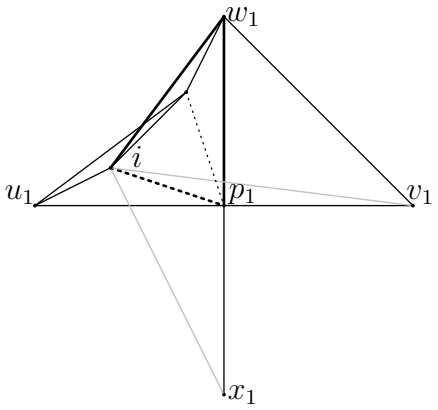
The axioms above are used for automating reasoning about geometric graphs and for proving the correctness of the CP-algorithm.

6.3.1 Decidability in the Proposed Axiom System

Analyzing the form of the axioms used in the proposed axiomatization (with the exception of the axioms describing the path along the convex hull), it turns out that they are all universally quantified formulae over a signature containing only constants and predicate symbols (thus no function symbol has an arity ≥ 1). Therefore, all the verification tasks, which do not refer



(**a**) The upper vertex of the highlighted triangle is $\mathsf{next}(i)$. i is located inside the triangle $\Delta u_1 p_1 \mathsf{next}(i)$



(**b**) The upper vertex of the highlighted triangle is $\mathsf{next}(i)$, which is located inside the triangle $\Delta p_1 w_1 i$

Figure 32: Axioms for the convex hull

to the paths along a convex hull, can be formulated as satisfiability tasks for formulae in the Bernays-Schönfinkel class.

Definition 32 (**The Bernays-Schönfinkel class**). *Let $\Sigma = \{c_1, \dots c_n\}$ be a signature consisting only of constants (it does not contain function symbols with arity ≥ 1).*
The Bernays-Schönfinkel class consists of all formulae of the form
$\exists x_1 \dots x_n \forall y_1 \dots \forall y_m F(x_1, \dots, x_n, y_1, \dots, y_m)$, where $n, m \geq 0$ and F is a quantifier-free formula containing only variables $x_1, \dots, x_n, y_1, \dots, y_m$.

Satisfiability of formulae in the Bernays-Schönfinkel class is decidable. The complexity of the satisfiability problem is in $NEXPTIME$ [46].

Description of paths along the convex hull of points inside of a triangle. McPeak and Necula investigate in [53], reasoning in pointer data structures. The language used has sorts p (pointer) and s (scalar). Sets Σ_p and Σ_s of pointer resp. scalar fields are modeled by functions of sort $\mathsf{p} \to \mathsf{p}$ and $\mathsf{p} \to \mathsf{s}$, respectively. A constant nil of sort p exists. The only predicate of sort p is equality; predicates of sort s can have any arity. The axioms considered in [53] are of the form

$$\forall p \;\; \mathcal{E} \vee \mathcal{C} \tag{66}$$

where $\mathcal{E}$ contains disjunctions of pointer equalities and $\mathcal{C}$ contains scalar constraints (i.e. sets of both positive and negative literals). It is assumed that for all terms of the form $f_1(f_2(\dots f_n(p)))$ occurring in the body of an axiom, the axiom also contains the disjunction $p = \mathsf{nil} \vee f_n(p) = \mathsf{nil} \vee \dots \vee f_2(\dots f_n(p)) = \mathsf{nil}$.[5]

[5]This nullable subterm property has the role of excluding null pointer errors.

In [53] a decision procedure for conjunctions of axioms of type (66) is proposed; in [38] a decision procedure based on instantiation of the universally quantified variables of sort p in axioms of type (66) is proposed.

The axioms **Y1**–**Y24** for the convex hull are all of type (66). Furthermore, the axioms also satisfy the nullable subterm condition listed above, since the considered graphs are assumed to be finite. The axioms which are needed for the verification conditions can therefore be represented as the problem of checking satisfiability tasks w.r.t. to the theory extension $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \{\mathbf{Y1}, \ldots, \mathbf{Y24}\}$ where the theory $\mathcal{T}_0$ axiomatized by $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ can be regarded as a theory of scalars. Therefore, satisfiability of ground instances w.r.t. $\mathcal{T}_1$ is decidable, and decision procedures based on instantiation exist.

6.3.2 Choosing the Proof Systems

Since the axioms can be formulated in first-order-logic, the theorem provers SPASS [78] and Z3 [23] were considered for solving the proof tasks. The advantage of Z3 is that for formulae in the Bernays-Schönfinkel class Z3 can be used as a decision procedure [63], whereas SPASS might not terminate. Since the axioms (with the exception of the modelling of the paths along the convex hull) are in the Bernays-Schönfinkel class, Z3 is prefered against SPASS. In addition, instantiation-based decision procedures exist for the type of list structures used for representing the considered paths along the convex hulls of points [53]. Since Z3 is an instantiation-based SMT-prover, it performs the necessary instances during the proof process.

A further method that was considered for solving the proof tasks, was a method for complete hierarchical reasoning in local theory extensions, proposed in [69, 38, 40], because the properties of geometric graphs were defined hierarchically. However, it was not possible to use the prover H-PILoT [39], because in the current implementation the hierarchical reduction in the system can only be used for chains of theory extensions in which for every extension in this chain the free variables occur below extension symbols. This is not the case in the proposed axiomatization, since the properties of new predicate symbols used variables, which do not occur below an extension function. In axiom **F4** the variable w does not occur below F and in the axioms **F5** and **F6** the variables w and x do not occur below F.

6.3.3 Automated Proof of Properties of Geometric Graphs

In a first step, the proposed axiomatization of plane geometry and the axioms of geometric graphs was considered. The following properties were automatically proven:

Independence of axioms: The axioms in $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ are all independent. However, **A5** can be derived from **A1-4** and **A6** if there exist at least 5 pairwise distinct vertices. The tests can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/Independence-Consistence.

Properties for left The predicate left is defined such that some properties (e.g. u is itself not in the left half-plane defined by the line passing through u and v) follow from the axiomatization for left. Therefore no axioms for the according properties, that already follow from **A1-6** have to be added in the axioms defining left An overview of properties derived by the axioms **A1-6** can be found in Appendix 9.3.1. The tests can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/Left

Properties for inside The predicate inside is defined such that it is transitive and for two distinct points d and e located inside the triangle Δabc, e is also located inside a triangle formed by a, b and d. The tests can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/Inside.

Transitivity Property (T11): It is proven that the axioms AxGeom entail:

$$\begin{aligned} &\mathsf{inside}(a,b,c,d) \wedge \mathsf{inside}(a,b,d,e) \\ \rightarrow &\mathsf{inside}(a,b,c,e) \end{aligned}$$

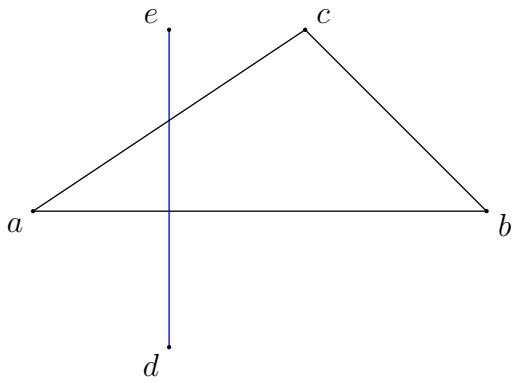
Smaller Triangle Property (T14): It is proven that the axioms AxGeom entail:

$$\begin{aligned} &\mathsf{inside}(a,b,c,d) \wedge \mathsf{inside}(a,b,c,e) \wedge d \neq e \\ \rightarrow &\mathsf{inside}(a,b,d,e) \vee \mathsf{inside}(b,c,d,e) \vee \mathsf{inside}(c,a,d,e) \end{aligned}$$

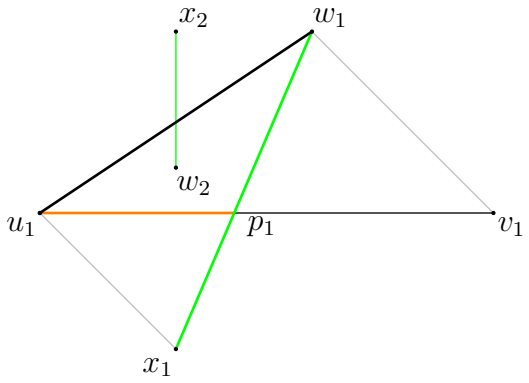
Relation to Hilbert's axiom system From the axioms defining the predicate left some of the axioms of Hilbert's axiom derived. Hilbert's axioms 2.1 and 2.3 follow immediately from the definition of the predicate left. Hilbert's axioms 2.2 contains an existential quantifier and is not required to hold for axiomatizing finite graphs. The axiom 2.4 in Hilbert's axiom system is also known as Pasch's axiom and states that if a line intersects one side of a triangle then it intersects exactly one other side. Pasch's axiom can be derived from the axioms AxGeom. For this 5 points a, b, c, d and e are considered, such that a, b and c are pairwise distinct, $c \neq d$ and $c \neq e$. Furthermore, neither c nor d are located inside a triangle formed by a, b and c. If the segments ab and de intersect, then de also intersects with ac or bc (see Fig. 33(**a**)). The tests, concerning the relation between AxGeom and Hilbert's axioms can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/Hilbert.

Pasch's axiom: It is proven that the axioms AxGeom entail:

$$
\begin{aligned}
& a \neq b \wedge a \neq c \wedge b \neq c \wedge c \neq d \wedge c \neq e \\
& \wedge \neg \mathsf{inside}(a, b, c, d) \wedge \neg \mathsf{inside}(a, b, c, e) \\
& \wedge \neg \mathsf{inside}(b, a, c, d) \wedge \neg \mathsf{inside}(b, a, c, e) \\
& \wedge \mathsf{intersection}(a, b, d, e) \\
& \rightarrow \mathsf{intersection}(a, c, d, e) \\
& \vee \mathsf{intersection}(b, c, d, e)
\end{aligned}
$$



(**a**) The segment de is intersecting the segments ab and ac



(**b**) The edge w_1x_1 is the edge in F with intersection point p_1 closest to u_1 among all edges in F intersecting u_1v_1

Figure 33: Illustration of Pasch's axiom in the general case and for the property of an edge w_1x_1 with intersection point closest to u_1

In Pasch's axiom the points a, b, c are required to be not collinear. Furthermore, the line intersecting with ab is required not to hit any of the points a, b and c. Those two conditions can be omitted in the derived implication, because in AxGeom, the predicates intersection and inside are also defined for collinear points. Therefore, the derived implication is even a bit more general than Pasch's axiom. One case that is considered in the implication, but not in Pasch's axiom, is that c is between d and e. In this case the segment, de is intersecting with all of the segments ab, ac and bc. However, the possibility that de is intersecting with ac and bc can be ruled out, by adding the condition that c is not between d and e. For the special case that b is between a and c an edge de with $c \neq d$ and $c \neq e$ intersecting with ab is also intersecting with ac.

closestF Pasch's axiom can also be used to derive conditions for the intersection of two edges u_1v_1 and w_1x_1, with intersection point p_1, where w_1x_1 is an edge in F and chosen such that it is the edge in F with intersection point closest to u_1[6] among all edges in F intersecting with u_1v_1. For the proofs in **closestF Step1** (see Fig. 33(**b**)) it is assumed that an edge w_2x_2 in F is deleting the edge u_1w_1. It is proven that the edge w_1x_1 can only be the edge in F with intersection point closest to u_1 among all edges in F intersecting with u_1v_1 if the following conditions are satisfied:

- w_2 or x_2 are located inside a triangle formed by u_1, w_1 and p_1 (**closestF Step1a**)
- w_2x_2 is not intersecting u_1v_1 (**closestF Step1b**)
- The vertex w_2 is not the vertex x_1 (**closestF Step1c**)

For the proofs in **closestF Step2** it is assumed that an edge w_2x_2 in F intersects u_1p_1. This edge would contradict the property that w_1x_1 was chosen as the edge in F with intersection point closest to u_1 among all edges in F intersecting with u_1v_1. In **closestF Step2a** it is proven that one of the vertices w_2 or x_2 is located inside a triangle formed by u_1, p_1, w_1, the edges u_1w_1 and w_2x_2 intersect or one of the vertices w_2 or x_2 is w_1. In **closestF Step2b** the case that $w_2 = w_1$ is considered and it is proven that the vertex x_2 is located inside a triangle formed by u_1, p_1, w_1, the edges u_1x_1 and w_1x_2 intersect or $x_2 = x_1$. In the case that $x_2 = x_1$ the edge w_2x_2 is the edge w_1x_1. The tests for the proof tasks can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/closestF.

closestF Step1a: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$
\begin{aligned}
&\mathsf{left}(u_1, v_1, p_1) \wedge \mathsf{left}(v_1, u_1, p_1) \wedge \mathsf{left}(w_1, x_1, p_1) \wedge \mathsf{left}(x_1, w_1, p_1) \\
&\wedge \mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge F(w_1, x_1) \\
&\wedge \mathsf{deleting}(u_1, w_1, w_2, x_2) \wedge F(w_2, x_2) \\
&\rightarrow \mathsf{inside}(u_1, p_1, w_1, w_2) \vee \mathsf{inside}(u_1, p_1, w_1, x_2) \\
&\vee \mathsf{inside}(u_1, w_1, p_1, w_2) \vee \mathsf{inside}(u_1, w_1, p_1, x_2) \\
&\vee \mathsf{intersection}(u_1, p_1, w_2, x_2)
\end{aligned}
$$

The tests of the remaining steps are similar and listed in Appendix 9.4.1.

For the proof tasks of the correctness proof the intersection point of two edges is not included as a separate point in the tests in order to reduce the running time. In the corresponding proof tasks, the condition that an edge in F is the edge with intersection point closest to one of the vertices of another edge is expressed by using the derived conditions instead.

[6]The case considering v_1 can be derived analogous by switching the labeling of u_1 and v_1

Properties for graphs in RCG Some properties of geometric graphs $G = (V, E)$ in the class RCG which are useful in the correctness proof of the CP-algorithm are proven. For $u, v, w \in V$, such that $uv, vw, wu \in E$ it follows from the coexistence property, that the vertices located inside the triangle Δuvw are forming a clique.

Moreover, for an edge $uv \in E$ and a vertex $w \in V$ located on the segment uv it follows from the redundancy and coexistence property that also $uw \in E$ and $vw \in E$. In order to prove this it is assumed that there exists a vertex $x \in V$, such that the $wx \in E$. The corresponding tests can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/RCG.

Clique Property: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{C1}\}$ entail:

$$\begin{aligned} & E(a,b) \wedge E(b,c) \wedge E(c,a) \\ \wedge & \mathsf{inside}(a,b,c,d) \wedge V(d) \\ \wedge & \mathsf{inside}(a,b,c,e) \wedge V(e) \wedge d \neq e \\ \rightarrow & E(d,e) \end{aligned}$$

Colinearity Property: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$\begin{aligned} & E(u,v) \wedge E(w,x) \wedge \mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \\ \rightarrow & E(u,w) \wedge E(v,w) \end{aligned}$$

7 Correctness Proof of the CP-algorithm

The following correctness proof of the CP-algorithm (partially already published in [14]) is based on a manually derived proof in [11]. The proof in [11] does not contain all details for all proof steps. In this section the proof is considered in full details with all relevant case distinctions.

7.1 Idea of the Correctness Proof

The CP-algorithm starts with a finite input graph $G = (V, E)$, which is connected and satisfies the redundancy and the coexistence property and constructs a connected and plane output graph $G' = (V, F)$. In order to prove the correctness of the CP-algorithm it has to be proven that the algorithm terminates, that the output graph is plane and that the output graph is connected.

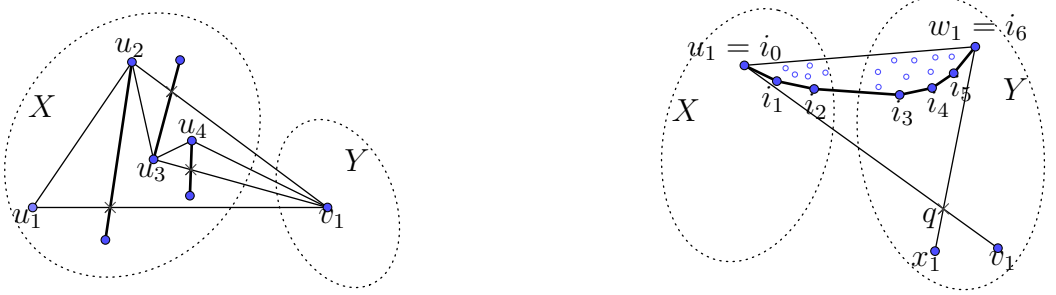
Termination: Since the graph G is finite and in every iteration of the while loop at least one edge is removed from the finite set of edges W, the algorithm terminates.

Planarity: From the formulation of the algorithm it follows that for an edge uv, which is added to F, all intersecting edges are removed from the working set W. Therefore, at any time W only contains edges which are not intersecting with edges so far added to F. This means that only those edges can be added to F, which are not intersecting with edges that are already in F. This leads to a plane output graph.

Connectivity: The proof is given by contradiction. It is assumed that after the termination of the algorithm, the resulting graph $G' = (V, F)$ has at least two components which are not connected. Let k be the number of components of the graph G', X be one of the components and Y be the induced subgraph of $G' \setminus X$, which contains the other $k - 1$ components. It is proven that a path with edges in F connecting a vertex in X and a vertex in Y exists, which is a contradiction to the assumption that X and Y are not connected in G'. Therefore, the number of components has to be one, which means that the graph is connected. For proving the connectivity of G', the following properties are needed. Each of the properties corresponds to one lemma and the correctness of the lemmas are verified in the following subsections.

- For every edge uv in $E \setminus F$ there exists an edge wx in F that intersects with uv **(Lemma 20)**.
- There exists an edge uv in E with $u \in X$ and $v \in Y$, such that all edges in F intersecting with uv are located in Y **(Lemma 21)**. (Example: the edge u_4v_1 in Fig. 34(**a**) which is resulting from edge sequence construction $u_1v_1,\ u_2v_1, u_3v_1, u_4v_1$ described in Section 7.3).
- For an edge uv and the intersecting edge $wx \in F$ with the redundancy edges wu and wv, and intersection point q closest to u among all edges in F, which intersect with uv,

the finest path P from u to w on the convex hull of the triangle Δuqw contains only edges in F **(Lemma 22)**. (Example: the path $i_0 i_1 \dots i_6$ from u_1 to w_1 in Fig. 34(**b**), constructed for the edge u_1v_1 and the intersecting edge $w_1x_1 \in F$).



(**a**) Illustration of a sequence of edges connecting X and Y.

(**b**) Illustration of the path on the convex hull of the triangle $\Delta u_1 q w_1$ connecting X and Y.

Figure 34: Connecting paths from X to Y.

Combining the results of the Lemmas 20–22 leads to the connectivity of G' with the following argumentation:

Since the original graph $G = (V, E)$ before applying the CP-algorithm was connected, there has to exist an edge u_1v_1 in E but not in F with $u_1 \in X$ and $v_1 \in Y$. Lemma 20 ensures that u_1v_1 is intersected by an edge w_1x_1 in F and by Lemma 21 u_1v_1 can be chosen such that all intersecting edges in F are located in Y. Lemma 22 ensures now – for the intersecting edge $w_1x_1 \in F$ (located in Y) with redundancy edges w_1u_1 and w_1v_1 and intersection point q closest to u_1 – the existence of a path P from u_1 to w_1 with all edges in F (see Fig. 34(**b**)). This path is connecting u_1 and w_1 and therefore connects X and Y in G'. This is a contradiction to the assumption that the components X and Y are not connected in G'. With the same argumentation all remaining components can be connected until the graph is connected. Therefore, the following theorem holds.

Theorem 10. *The graph G' is a connected plane spanning subgraph of G after the termination of the CP-algorithm.*

The proofs of Lemma 20, Lemma 21 and Lemma 22 follow manually derived ideas. All steps of those manually derived proofs are automatically verified by using Z3. While the properties that are proven in Lemma 20, Lemma 21 and Lemma 22 refer to the existence of certain edges resp. paths connecting vertices, all proof steps are formulated in a way, that avoids quantifier alternations. Formulating the proof steps without using existential quantification makes it easier for Z3 to verify the correctness of the proof steps. Therefore, one cannot handle the properties to be proven in the lemmas to Z3 directly, but has to divide the proof in many small steps manually, before using Z3. Putting all proof steps together, completes the proofs

of the lemmas by inductively repeating the proof steps in the general manually derived proof concept. Since there exists a high number of case distinctions and some of the proof tasks are quite complicated, computer support is needed to verify the correctness of the proof steps in full detail.

7.2 Removed Edges are Intersected by Selected Edges (Lemma 20)

The first task in the correctness proof of the CP-algorithm is to regard the edges, which are removed from the working set W, without being added to the result set F. Regarding the resulting graph from the CP-algorithm, it is not clear why an edge uv was deleted, i.e. removed from W, without being added to F. It can either be that an intersecting edge was added to F before or that the CP-condition is not satisfied for an intersecting edge. A crucial question is whether it is possible that also all edges intersecting with uv are themselves removed from the working set without being added to F. In order to answer this question it is investigated, if it is possible to construct an acyclic redundancy sequence of edges intersecting with uv. In such a redundancy sequence every edge, except the first edge, is a redundancy edge of an intersection, where the previous edge of the sequence is the redundant edge, which is deleted by the intersecting edge. If such a sequence exists, the edge at the end of the sequence is not deleted by another edge and therefore in F.
Constructing such a redundancy sequence is a very complicated task for the following reasons:

- An edge is usually deleted by multiple edges and it is not obvious which of the deleting edges should be chosen to construct the redundancy sequence
- It is not obvious which properties of such a sequence can be proven, that guarantee that the sequence is acyclic
- Such a sequence can be arbitrary long and therefore a suitable choice for case distinctions is necessary, such that all possible sequences are covered, with a limited number of cases

It turns out that it is possible to construct such an acyclic redundancy sequence and therefore the following lemma holds:

Lemma 20. *For every edge uv that is not in F, an edge wx in F exists that intersects with uv.*

Proof. Let $e_0 = u_1v_1$ be an edge in E that is not in F. If there exists no edge intersecting with u_1v_1, then u_1v_1 would be in F. So it can be assumed that there exists at least one edge in E intersecting with u_1v_1. A sequence of edges in E intersecting u_1v_1 is constructed in the following way: The first edge of the sequence is $d_1 = w_1x_1$ that deletes, and therefore also intersects u_1v_1. If $w_1x_1 \in F$ the proof is completed. In order to obtain a contradiction it is assumed that all edges intersecting with u_1v_1 are not in F.

This is expressed by adding the axiom **P1**: $\forall w, x : \mathsf{intersection}(u_1, v_1, w, x) \rightarrow \neg \mathsf{F}(w, x)$

According to Paschs axiom for colinear vertices, for a vertex w_1 located on the edge u_1v_1, an edge yz intersecting u_1w_1 is also intersecting u_1v_1 if yz is not the edge u_1v_1. Analogously an edge yz intersecting v_1w_1 is also intersecting u_1v_1 if yz is not the edge u_1v_1. It is therefore assumed that there are no other vertices located on the edge u_1v_1.
This is expressed by adding the axiom **P2**: $\forall x : \mathsf{V}(x) \rightarrow \neg \mathsf{left}(u_1, v_1, x) \vee \neg \mathsf{left}(v_1, u_1, x)$

Starting with the edge w_1x_1, a sequence $e_1, \ldots, e_{i-1}$ of *eligible edges* in E intersecting with u_1v_1 is constructed. For every edge of the sequence a *deleting edge* d_i in E is chosen, that deletes e_{i-1}, i.e. d_i is an edge, which prohibits that e_{i-1} is in F. Since the first edge w_1x_1 is not only intersecting, but also deleting the edge u_1v_1, it holds that $d_1 = e_1 = w_1x_1$. For the further edges of the sequence ($i \geq 2$) let e_i be a redundancy edge of the intersection between e_{i-1} and d_i, such that e_i intersects with u_1v_1. In the corresponding proof steps it is proven that d_i can always be chosen such that one of the redundancy edges, resulting from the intersection of e_{i-1} and d_i, is intersecting with u_1v_1. The sequence of eligible edges is therefore a redundancy sequence, where all edges of this sequence are intersecting with u_1v_1. Since no vertices are located on the edge u_1v_1, for every edge e_i of the sequence one vertex of e_i is located on the same side of u_1v_1 as w_1 and the other vertex is located on the same side of u_1v_1 as x_1. Therefore, for every edge e_i of the sequence the following terminology is used: $e_i = y_{f(i)}z_{g(i)}$, where $y_{f(i)}$ is the vertex of the edge e_i located on the same side of u_1v_1 as w_1 and $z_{g(i)}$ is the vertex of e_i located on the same side of u_1v_1 as x_1. The first edge of the sequence is therefore $e_1 = w_1x_1$ with $y_{f(1)} = w_1$ and $z_{g(1)} = x_1$. For the further eligible edges the definition of the labeling for $f(i) = m_n$ and $g(i) = r_s$ is described in the following and illustrated in Fig. 35.

Since for $i \geq 2$ the edge e_i is always a redundancy edge of the intersection between the edges $e_{i-1} = y_{f(i-1)}z_{g(i-1)}$ and $d_i = w_ix_i$, either $y_{f(i)} = y_{f(i-1)}$ or $z_{g(i)} = z_{g(i-1)}$ holds for every edge e_i (with $i \geq 2$) of the sequence. The edge e_i is therefore either $y_{f(i-1)}w_i$ or $w_iz_{g(i-1)}$. The vertex w_i is termed an *eligible vertex*. If $e_i = w_iz_{g(i-1)}$, then the edge $d_i = w_ix_i$ is called an edge of *type* Y, because $y_{f(i)} = w_i$. On the other hand if $e_i = y_{f(i-1)}w_i$, then the edge $d_i = w_ix_i$ is called an edge of *type* Z, because $z_{g(i)} = w_i$. A deleting edge of type Y is denoted as $d_i(Y)$, while a deleting edge of type Z is denoted as $d_i(Z)$.

In Fig. 35 the way the deleting edges are chosen is illustrated. The decision $d_i(Y)$ represents the question, whether there exists a deleting edge of type Y, which deletes the edge e_{i-1}. Analogously the decision $d_i(Z)$ represents the question, whether there exists a deleting edge of type Z, which deletes the edge e_{i-1}.

For the first edge $e_1 = w_1x_1$ intersecting with u_1v_1, the edge deleting e_1 is d_2, therefore the construction of the sequence starts with $i = 2$. Furthermore, the start values of m and n are set to $m = 1$ and $n = 1$, such that $y_{1_1} = w_1$. The start values of r and s are set to

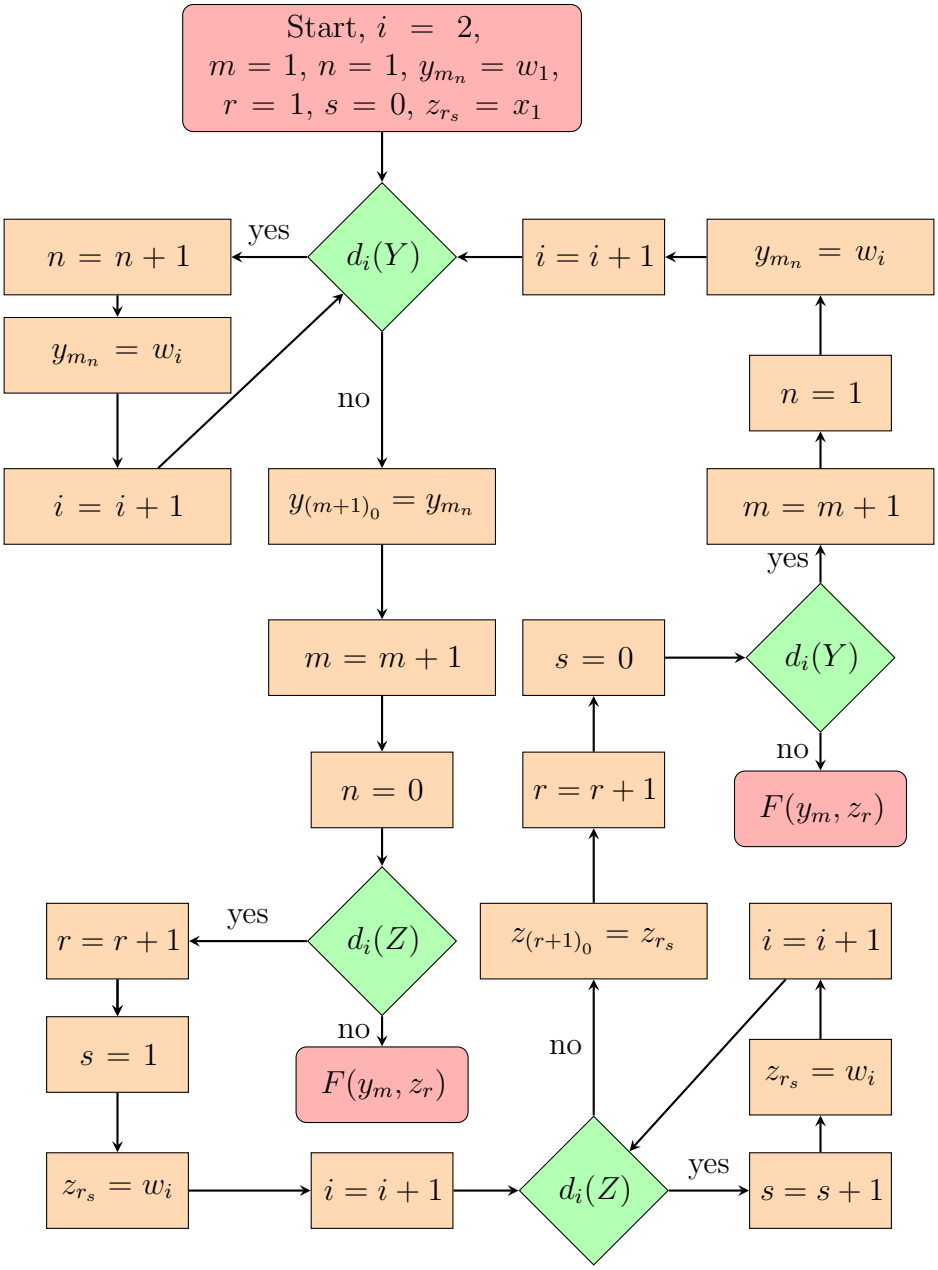


Figure 35: Updating the values of $g(i)$ and $f(i)$

$r = 1$ and $s = 0$, such that $z_{1_0} = x_1$. The sequence is constructed by considering initially only deleting edges of type Y, until no type Y edge is available anymore, which deletes an edge e_i. As long as a deleting edge of type Y is available the value of n is increased by 1 in every iteration step. Once an edge e_i is reached, where no deleting edge of type Y is available anymore, $f(i)$ is set to $(m+1)_0$ and the edge e_i is termed a *switching edge* the corresponding vertex w_i is termed a *switching vertex*. If there is also no edge of type Z available that deletes the edge e_i, then e_i is in F and the proof is completed. If there exists at least one type Z edge deleting e_i, the value of r is increased by 1 and the value of s is set to 1. The sequence

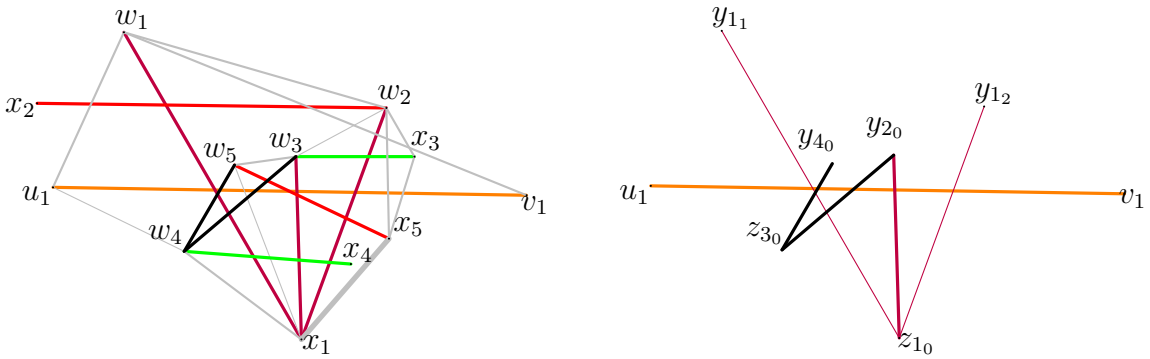
is continued using only deleting edges of type Z and increasing the value of s by 1 in every iteration step, until an edge e_j is reached, where no type Z edge deleting e_j is available. The edge e_j is also a switching edge. The value of $g(j)$ is set to $(r+1)_0$. If there exists also no edge of type Y that deletes the edge e_j, then e_j is in F and the proof is completed. If there exists at least one type Y edge deleting e_j, the value of m is increased by 1 and the value of n is set to 1. The construction of the sequence is continued using again only deleting edges of type Y and updating the value of n by 1 in every iteration step, until an edge e_k is reached, where no type Y edge deleting e_k is available. This process of alternating considering only deleting edges of type Y and only deleting edges of type Z is continued until an edge e_s is reached, where no edge is available that deletes e_s or e_s is an edge that was already part of the sequence and the sequence would therefore be cyclic. An example for a sequence of eligible and deleting edges with the according labeling of the vertices is illustrated in Fig. 36 and described in the following.

The first edge of this sequence is $e_1 = d_1 = w_1x_1$. For the sequence of eligible edges the labeling of this edge e_1 is chosen as $y_{1_1} = w_1$ and $z_{1_0} = x_1$. For e_1 there exists an deleting edge $d_2 = w_2x_2$ of type Y. The redundancy edge $e_2 = w_2x_1$ of the intersection between e_1 and d_2 is intersecting with u_1v_1. The labeling of e_2 is chosen as $y_{1_2} = w_2$ and $z_{1_0} = x_1$.

Next the edge $d_3 = w_3x_3$ of type Y is deleting e_2. The redundancy edge $e_3 = w_3x_1$ of the intersection between e_2 and d_3 is intersecting with u_1v_1. The labeling of e_3 is chosen as $y_{1_3} = w_3$ and $z_{1_0} = x_1$. There is no edge of type Y available deleting e_3, therefore the labeling of the vertex w_3 is updated to y_{2_0}. The edge $y_{2_0}z_{1_0}$ is a switching edge and the vertices $w_3 = y_{2_0}$ and $x_1 = z_{1_0}$ are switching vertices.

For the edge e_3 there exists a deleting edge $d_4 = w_4x_4$ of type Z. The redundancy edge $e_4 = w_3w_4$ of the intersection between e_3 and d_4 is intersecting with u_1v_1. The labeling of e_4 is chosen as $y_{2_0} = w_3$ and $z_{2_1} = w_4$. There is no edge of type Z available deleting e_4, therefore the labeling of the vertex w_4 is updated to z_{3_0}. The edge $y_{2_0}z_{3_0}$ is a switching edge and the vertices $w_3 = y_{2_0}$ and $w_4 = z_{3_0}$ are switching vertices.

For the edge e_4 there exists a deleting edge $d_5 = w_5x_5$ of type Y. The redundancy edge $e_5 = w_5w_4$ of the intersection between e_4 and d_5 is intersecting with u_1v_1. The labeling of e_5 is chosen as $y_{3_1} = w_5$ and $z_{3_0} = w_4$. There is no edge of type Y available deleting e_4, therefore the labeling of the vertex w_5 is updated to y_{4_0}. The edge $y_{4_0}z_{3_0}$ is a switching edge and the vertices $w_5 = y_{4_0}$ and $w_4 = z_{3_0}$ are switching vertices. There is also no edge of type Z available deleting e_5. Therefore, the edge $y_{4_0}z_{3_0}$ is an edge in F intersecting with u_1v_1.



(**a**) A sequence of eligible and deleting edges (**b**) The sequence of eligible edges

Figure 36: Illustration of the labeling for the eligible edges. Some edges, required to satisfy the coexistence property, are omitted due to simpler illustration

The labeling of $f(i) = m_n$ and $g(i) = r_s$ is chosen according to Fig. 35 and has the following recursive definition:

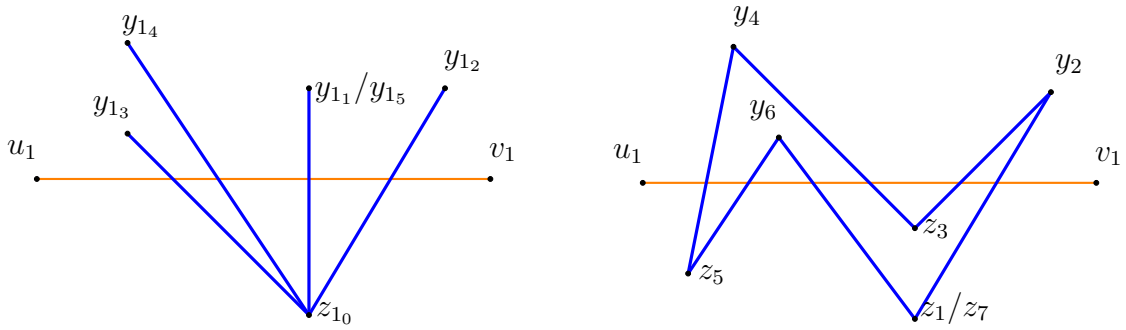
$$
\begin{aligned}
&f : \mathbb{N} \mapsto \mathbb{N} \times \mathbb{N}_0 \\
&f(1) = 1_1 \\
&f(i) = (m+1)_1 && \text{if } z_{g(i)} = z_{g(i-1)} \text{ and } n = 0 \\
&f(i) = m_{n+1} && \text{if } z_{g(i)} = z_{g(i-1)} \text{ and } n \neq 0 \\
&f(i) = m_n && \text{if } y_{f(i)} = y_{f(i-1)} \text{ and } n = 0 \\
&f(i) = (m+1)_0 && \text{if } y_{f(i)} = y_{f(i-1)} \text{ and } n \neq 0
\end{aligned}
$$

$$
\begin{aligned}
&g : \mathbb{N} \mapsto \mathbb{N} \times \mathbb{N}_0 \\
&g(1) = 1_0 \\
&g(i) = (r+1)_1 && \text{if } y_{f(i)} = y_{f(i-1)} \text{ and } s = 0 \\
&g(i) = r_{s+1} && \text{if } y_{f(i)} = y_{f(i-1)} \text{ and } s \neq 0 \\
&g(i) = r_s && \text{if } z_{g(i)} = z_{g(i-1)} \text{ and } s = 0 \\
&g(i) = (r+1)_0 && \text{if } z_{g(i)} = z_{g(i-1)} \text{ and } s \neq 0
\end{aligned}
$$

For simplification the indices n and s are omitted in the notation if they are not explicitly required and the simplified labeling y_m and y_r is used instead.

A cyclic sequence among the eligible edges can occur if all deleting edges leading to the cyclic sequence are of type Y (see Fig 37(**a**)), all deleting edges are of type Z or among the switching

edges (see Fig 37(**b**)).



(**a**) A cyclic sequence formed only by deleting edges of type Y

(**b**) A cyclic sequence formed by deleting edges of type Y and Z

Figure 37: Two possibilities of a cyclic sequence

Furthermore, it has to be proven that all eligible edges are intersecting with u_1v_1. In order to prove that all eligible edges are intersecting with u_1v_1 and that no cycle occurs among the eligible edges the proof of Lemma 20 is divided in the following 3 parts:

Part1: First it is proven that all eligible edges are intersecting with u_1v_1. For this conditions are derived, when an eligible edge is intersecting with u_1v_1 and in the following proof parts it is proven that always one of those conditions is satisfied.

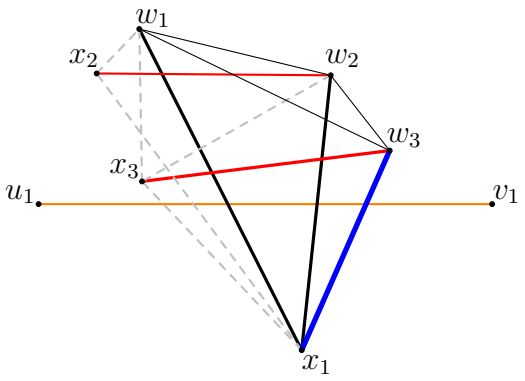
Part2: For a possible redundancy cycle, first the case that the sequence is constructed only by deleting edges of type Y or only by deleting edges of type Z is considered. It is proven that it is always possible to choose the deleting edges, such that for any r and $s' > s$ no redundancy cycle of the eligible edges occurs, where $y_{r_s} = y_{r_{s'}}$. Analogously it is always possible to choose the deleting edges, such that for any m and $n' > n$ no redundancy cycle of the eligible edges occurs, where $z_{m_n} = z_{m_{n'}}$.

Part3: Finally the case that the sequence is constructed by deleting edges of type Y and Z is considered. It is proven that it is always possible to choose the deleting edges, such that for any $r' > r$ and $m' > m$ no redundancy cycle of the eligible edges occurs, where $y_r = y_{r'}$ and $z_m = z_{m'}$

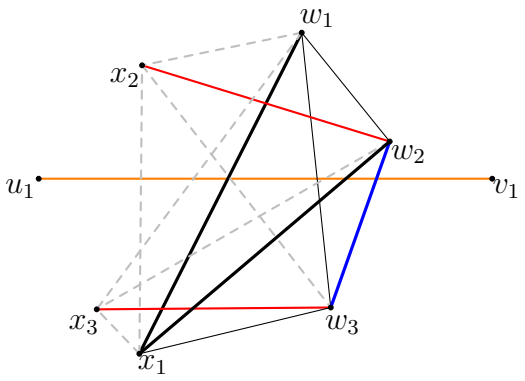
In order to cover all possible cases leading to a redundancy cycle, such a sequence S starting with e_1 and leading to a redundancy cycle is considered in the following. Let e_a be the first edge that occurs twice in the sequence, i.e. the sequence contains the edges e_1, e_2, ..., e_{a-1}, e_a, e_{a+1},..., e_a. If e_a is not only intersecting, but also deleting u_1v_1, then e_a can be considered as the edge e_1. If e_a is only intersecting, but not deleting u_1v_1, then also the part of the sequence from e_1 to e_a and then the redundancy cycle itself have to be considered in order to ensure that all edges of the redundancy cycle intersect with u_1v_1.

In order to prove that all edges of the sequence are intersecting with u_1v_1, a subsequence S' of S of shorter length is constructed in the following way: Let e_i and e_j with $j \geq i+2$ be part of the sequence and let d'_{i+1} be one of the available deleting edges for deleting e_i. If the resulting eligible edge of the intersection between e_i and d'_{i+1} is e_j, then d'_{i+1} is chosen instead of d_{i+1}, which results in a subsequence of shorter length. An example is illustrated in Fig. 38(**a**), where the edge w_3x_3 can be chosen as d'_2. However, the original sequence S, without choosing alternative deleting edges in order to shorten the length of the sequence, is still regarded in parallel, in order to detect redundancy cycles. Once an edge e_a is reached that occurs twice in the sequence it remains to consider only the subsequence of edges forming the redundancy cycle, because all of the edges in this cycle where already considered in a sequence S' and are therefore intersecting with u_1v_1. Also for edges forming the redundancy cycle the construction of a shorter subsequence is applied. Without considering the sequences S' and S separately a redundancy cycle might not be detected.

It has to be noted that constructing a subsequence in this way is only possible if d'_{i+1} is of the same type as d_{i+1}. If the condition that choosing the edge d'_{i+1} has to result in the eligible edge e_j is dropped, then only a shorter sequence, but not necessarily a subsequence is constructed. In a shorter sequence, that is not a subsequence, the switching edges would not be fixed anymore. However, without fixing the switching edges, the edge d_{i+1} might again delete the edge e'_{i+1} obtained by choosing d'_{i+1} as deleting edge for e_i. Since d_{i+1} is also deleting e_i it would again be chosen as alternative edge deleting e_i. This leads to the same sequence as considered before, so it cannot be proven that a termination for the construction of the sequence is obtained in this case. An example of this case is illustrated in Fig. 38(**b**), where w_3x_3 is deleting w_1x_1 and w_2x_2 is deleting w_1w_3. However, w_3x_3 cannot be chosen as d'_2, because it is a deleting edge of type Z



(**a**) The edge w_3x_3 can be chosen as alternative deleting edge d'_2 instead of w_2x_2

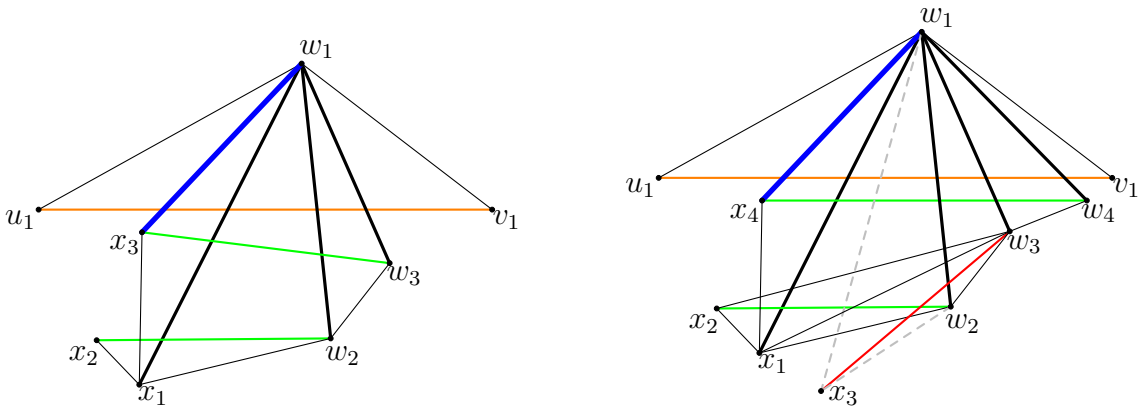


(**b**) No shorter sequence can be constructed, because w_2x_2 is of type Y and w_3x_3 of type Z

Figure 38: Constructing a shorter sequence is only possible for deleting edges of the same type

For choosing the deleting edges it turns out that in general there is no trivial rule for an

optimal choice to construct a shortest sequence. Therefore, the concept of finding a shorter subsequence has to be applied after every construction step of the sequence. However, among the deleting edges in F the situation is clearer, because they cannot intersect. An edge in F can only be chosen as deleting edge, if it is the edge with intersection point closest to u_1v_1 among all deleting edges in F for the considered type Y resp. Z of deleting edges available at this step of the sequence (see Fig. 39). Such an edge in F is determined as a *closestF edge*. Without this restriction, the following situation could occur, where w_2x_2 is deleting w_1x_1 and w_3x_3 is deleting w_1w_2, but only x_3w_3 is deleting w_1x_1. Furthermore, neither x_2w_2 is deleting w_1x_1, nor x_3w_3 is deleting w_1w_2. The problem in this case is that for edges $w_ix_i \in F$, once w_ix_i can be considered as the deleting edge and once x_iw_i. This leads to the problem that for a cyclic sequence containing the edges w_1x_1, w_1w_2 and w_1w_3 no shorter subsequence can be constructed by choosing x_3w_3 instead of w_2x_2 as deleting edge for w_1x_1, because w_1x_3 is not part of this cyclic sequence. The edges w_2x_2 and w_3x_3 are termed as *conflicting deleting edges*. In **orderF Step1** it is proven that with the restriction that only the edge in F with intersection point closest to u_1v_1 among all available edges of type Y resp. type Z can be chosen as deleting edge, the occurrence of conflicting deleting edges can be avoided (see Fig. 39(**a**)). A similar result is derived in **orderF Step2** for an edge $d_4 = w_4x_4$, where x_4w_4 is deleting w_1x_1 (see Fig. 39(**b**)). This restriction for choosing the deleting edges in F is assumed to hold for all steps of Part1-3.



(**a**) The edge x_3w_3 is deleting w_1x_1

(**b**) The edge x_4w_4 is deleting w_1x_1

Figure 39: Illustration of two possible cases for conflicting deleting edges in F

In the following implications ($\leftarrow$ Step x) means that the according condition can be assumed to hold, because the case considering the opposite condition was already considered in Step x. Furthermore, ($\rightarrow$ Step x) means that the according case is considered in Step x.

The tests of the following proof tasks can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/orderF.

orderF Step1: It is proven that conflicting deleting edges can only occur if both edges d_2 and d_3 are in F. For this it is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$
\begin{array}{rr}
\mathsf{deleting}(w_1, x_1, w_2, x_2) & (e_1 = w_1x_1, d_2 = w_2x_2) \\
\wedge \mathsf{deleting}(w_1, w_2, w_3, x_3) & (e_2 = w_1w_2, d_3 = w_3x_3) \\
\wedge \mathsf{deleting}(w_1, x_1, x_3, w_3) & (d_2^* = x_3w_3) \\
\rightarrow \mathsf{deleting}(w_1, x_1, w_3, x_3) & (d_2' = w_3x_3) \\
\vee \mathsf{deleting}(w_1, w_2, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Part2\ Step2)) \\
\vee \mathsf{F}(w_2, x_2) &
\end{array}
$$

orderF Step2: It is proven that conflicting deleting edges can only occur if both edges d_2 and d_4 are in F. For this it is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$
\begin{array}{rr}
\mathsf{deleting}(w_1, x_1, w_2, x_2) & (e_1 = w_1x_1, d_2 = w_2x_2) \\
\wedge \mathsf{deleting}(w_1, w_2, w_3, x_3) & (e_2 = w_1w_2, d_3 = w_3x_3) \\
\wedge \mathsf{deleting}(w_1, w_3, w_4, x_4) & (e_3 = w_1w_3, d_4 = w_4x_4) \\
\wedge \mathsf{deleting}(w_1, x_1, x_4, w_4) & (d_2^* = x_4w_4) \\
\rightarrow \mathsf{deleting}(w_1, x_1, w_3, x_3) & (d_2' = w_3x_3) \\
\vee \mathsf{deleting}(w_1, w_2, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Part2\ Step2)) \\
\vee \mathsf{deleting}(w_1, x_1, w_4, x_4) & (d_2' = w_4x_4) \\
\vee \mathsf{deleting}(w_1, w_3, x_4, w_4) & (d_{4b} = x_4w_4)(\rightarrow Part2\ Step2)) \\
\vee \mathsf{F}(w_2, x_2) &
\end{array}
$$

Another case that can occur for an edge $w_{i+1}x_{i+1}$ in F, is that $w_{i+1}x_{i+1}$ and $x_{i+1}w_{i+1}$ are both deleting an edge e_i. If $d_{2a} = w_2x_2$ and $d_{2b} = x_2w_2$ are both deleting w_1x_1, a redundancy cycle can only occur for choosing $d_2 = d_{2a}$ or $d_2 = d_{2b}$, but not in both cases. Therefore, d_2 can be chosen such that a redundancy cycle does not occur. The details of this case are considered in Part2 Step6. For $i \geq 2$ always a shorter subsequence can be obtained or one of the vertices w_{i+1} or x_{i+1} is located inside a triangle formed by vertices of previous edges in the sequence. The labeling of the edge d_{i+1} is chosen such that for the deleting edge $w_{i+1}x_{i+1}$ a subsequence of shorter length is obtained or w_{i+1} is located inside a triangle formed by the vertices of previous edges of the sequence. The details of this case are considered in Part1 Step3 resp. Part2 Step2.

The three parts of the proofs are now described in detail, starting with Part1, where it is proven that all eligible e_i edges are intersecting with u_1v_1

7.2.1 Proof of Part1: Intersection of the Eligible Edges with u_1v_1

This part of the proof considers, in 9 steps, all the cases that can occur in the redundancy sequence of eligible edges, starting with an edge w_1x_1 that deletes an edge u_1v_1. In each step it is proven that the currently considered eligible edge is intersecting with u_1v_1 or that a case occurs that is considered in one of the next steps. It is furthermore proven that it is not possible that one of the vertices u_1 or v_1 is located inside a triangle formed by the vertices of two eligible edges that have a vertex in common. This condition is necessary to prove the correctness of some limit cases. Since the proof tasks of each of the proof steps are quite similar, usually only one proof task per step is listed in the following. The remaining proof tasks are listed in Appendix 9.4.3. Afterwards, in Part2 (Section 7.2.2) and Part3 (Section 7.2.3) of the proof, it is proven that every eligible edge corresponds to one of the cases considered in this proof part. Therefore, all eligible edges can be assumed to intersect with u_1v_1 The test of the all the following proof tasks of Part1 can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/part1

Step1: (see Fig. 40) The first edge of the sequence of eligible edges is $e_1 = w_1x_1$. This edge also deletes u_1v_1, therefore $d_1 = e_1$. It is assumed that e_1 is not in F, because an intersecting edge w_2x_2 deletes w_1x_1, which means that the two redundancy edges w_1w_2 and x_1w_2 exist. It is proven in **Step1a** that one of the two redundancy edges intersects with u_1v_1. In the case that w_2x_1 is intersecting with u_1v_1, the edge w_2x_1 also deletes u_1v_1. in **Step1b** it is proven that u_1[7] is not located inside one of the triangles $\Delta w_1x_1w_2$ **(Step1b1)** or $\Delta x_1w_1w_2$ **(Step1b2)**.

(**a**) The edges u_1v_1 and w_1w_2 intersect. (**b**) The edges u_1v_1 and x_1w_2 intersect.

Figure 40: One of the edges w_1w_2 or x_1w_2 intersects with u_1v_1.

Step1a: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$\begin{aligned} &\mathsf{deleting}(u_1, v_1, w_1, x_1) && (e_0 = u_1v_1, e_1 = d_1 = w_1x_1) \\ \wedge&\mathsf{deleting}(w_1, x_1, w_2, x_2) && (e_1 = w_1x_1, d_2 = w_2x_2) \\ \rightarrow&\mathsf{deleting}(u_1, v_1, w_2, x_1) && (e_2 = w_2x_1 (\rightarrow \mathit{Step1})) \\ \vee&\mathsf{intersection}(u_1, v_1, w_1, w_2) && (e_2 = w_1w_2 (\rightarrow \mathit{Step2})) \end{aligned}$$

[7]The proof for v_1 can be derived by switching the labeling of u_1 and v_1

Step1b1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$\begin{aligned} &\mathsf{deleting}(u_1, v_1, w_1, x_1) && (e_0 = u_1v_1, e_1 = d_1 = w_1x_1) \\ \wedge&\mathsf{deleting}(w_1, x_1, w_2, x_2) && (e_1 = w_1x_1, d_2 = w_2x_2) \\ \wedge&\mathsf{inside}(w_1, x_1, w_2, u_1) \\ \rightarrow&\bot \end{aligned}$$

Step1b2: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

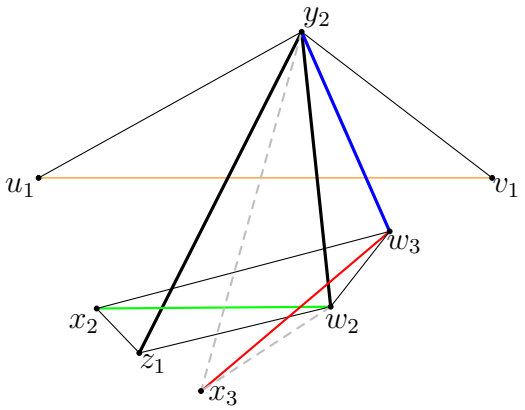
$$\begin{aligned} &\mathsf{deleting}(u_1, v_1, w_1, x_1) && (e_0 = u_1v_1, e_1 = d_1 = w_1x_1) \\ \wedge&\mathsf{deleting}(w_1, x_1, w_2, x_2) && (e_1 = w_1x_1, d_2 = w_2x_2) \\ \wedge&\mathsf{inside}(x_1, w_1, w_2, u_1) \\ \rightarrow&\bot \end{aligned}$$

If w_2x_1 is intersecting with u_1v_1, then the edge e_2 can be considered instead of w_1x_1 in Step1. In Part2 it is proven that the deleting edges of type Y can be chosen in a way, such that the resulting eligible edges with $z_{g(i)} = x_1$ are not forming a redundancy cycle. The case of an intersection between u_1v_1 and w_1w_2 is considered in Step2.

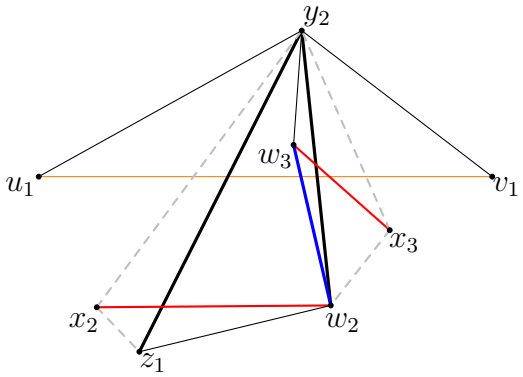
If the coexistence property is dropped, the implication of Step 1 does not hold anymore and a counterexample is constructed (see Sec. 7.4).

From Step1 it follows that all edges $y_{1_n}z_1$ are deleting u_1v_1. Therefore, also the switching edge y_2z_1 is deleting u_1v_1 and the edge y_2z_1 is considered as e_1 in the following steps. For this switching edge no deleting edge of type Y is available anymore. Therefore, for all edges wx deleting y_2z_1, the edge y_2w is intersecting u_1v_1.
This is expressed by adding the axiom:
Z1: $\forall w, x : \mathsf{deleting}(y_2, z_1, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_2, w)$

Step2: (see Fig. 41) For the intersection of e_1 and $d_2 = w_2x_2$ the redundancy edge $e_2 = y_2w_2$ is intersecting u_1v_1. In **Step2a1** an edge $d_3 = w_3x_3$ deleting e_2 is considered. Then one of the edges y_2w_3 or w_3w_2 is intersecting with u_1v_1 or the edge x_3w_3 is deleting the edge y_2w_2. In the remaining cases of **Step2a** it is proven that for all eligible vertices z_{2_s}, the edge $z_1z_{2_s}$ exists if no redundancy cycle occurs. In **Step2b** it is proven that u_1 is not located inside a triangle formed by w_3, z_1 and y_2, while in **Step2c** it is proven that u_1 is not located inside a triangle formed by w_3, y_2 and z_3.



(**a**) The edge y_2w_3 is intersecting with u_1v_1



(**b**) The edge w_3w_2 is intersecting with u_1v_1

Figure 41: Illustration of two possible cases in Step2

Step2a1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \mathbf{Z1}$ entail:

$$
\begin{array}{rll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3 = w_3x_3) \\
\wedge\neg & \mathsf{deleting}(y_2, z_1, x_3, w_3) & (\leftarrow orderF) \\
\rightarrow & \mathsf{intersection}(u_1, v_1, y_2, w_3) & (d_3(Z), e_3 = y_2w_3(\rightarrow Step2/4/6)) \\
\vee & \mathsf{intersection}(u_1, v_1, w_3, w_2) & (d_3(Y), e_3 = w_3w_2(\rightarrow Step4/8/9) \\
\vee & \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Step3))
\end{array}
$$

The case that x_3w_3 is also deleting y_2w_2 is considered in Step3. Otherwise, the edge e_3 is intersecting with u_1v_1.

Step3: In this step an eligible edge y_2z_2 is deleted by an edge w_3x_3 and also by x_3w_3. Then one of the edges y_2w_3, y_2x_3, w_3z_2 or x_3z_2 is intersecting u_1v_1 **(Step3a)**. The labeling of w_3 and x_3 can therefore be chosen such that y_2w_3 or w_3z_2 is intersecting with u_1v_1. In the tests of **Step3b** it is proven that u_1 and v_1 are not located inside a triangle formed by z_1, y_2 and w_3, while in the tests of **Step3c** it is proven that u_1 and v_1 are not located inside a triangle formed by y_2, w_2 and w_3.

Step3a: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup Z1$ entail:

$$
\begin{array}{lr}
\mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge\ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge\ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge\neg\mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge\neg\mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge\neg\mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge\neg\mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1/2) \\
\rightarrow\ \mathsf{intersection}(u_1, v_1, y_2, w_3) & (d_3(Z) = d_{3a}, e_3 = y_2w_3(\rightarrow Step2/4)) \\
\vee\ \mathsf{intersection}(u_1, v_1, y_2, x_3) & (d_3(Z) = d_{3b}, e_3 = y_2x_3(\rightarrow Step2/4)) \\
\vee\ \mathsf{intersection}(u_1, v_1, w_3, w_2) & (d_3(Y) = d_{3a}, e_3 = w_3w_2(\rightarrow Step4/8/9)) \\
\vee\ \mathsf{intersection}(u_1, v_1, x_3, w_2) & (d_3(Y) = d_{3b}, e_3 = x_3w_2(\rightarrow Step4/8/9))
\end{array}
$$

The proofs of Step3 hold for any eligible edge y_2z_2. Therefore, it can be assumed in the following steps of the proof that only w_ix_i, but not also x_iw_i deletes an eligible edge y_2z_2

Step4: In this step a vertex w_3 located inside a triangle formed by the vertices of two previous edges y_2z_1 and y_2w_2 of the sequence is regarded. In **Step4a** the vertex w_3 is located inside the triangle $\Delta z_1w_2y_2$, while in **Step4b** the vertex w_3 is located inside the triangle $\Delta w_2z_1y_2$. In **Step4a1** resp. **Step4b1** it is proven that y_2w_3 or w_2w_3 intersects with u_1v_1. In the remaining test of **Step4a** and **Step4b** it is proven that u_1 is not located inside a triangle formed by y_2, z_1 and w_3 resp. y_2, w_2 and w_3.

Step4a1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$
\begin{array}{lr}
\mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge\neg\mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2/3) \\
\wedge\neg\mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2/3) \\
\wedge\ \mathsf{inside}(w_2, z_1, y_2, w_3) & \\
\rightarrow\ \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2w_3(\rightarrow Step5/7)) \\
\vee\ \mathsf{intersection}(u_1, v_1, w_3, w_2) & (e_3 = w_3w_2(\rightarrow Step4/8/9))
\end{array}
$$

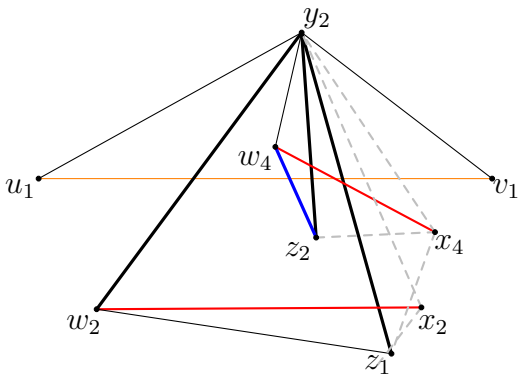
The proofs of Step4 can be adapted by changing the labeling of the vertices accordingly, for any eligible edges of the sequence, where a vertex is located inside a triangle formed by the vertices of two previous eligible edges, that have a common vertex.

Step5: (see Fig. 42) Now the case that $e_3 = y_2z_2$, where the vertex z_2 is located inside a triangle formed by y_2, z_1 and w_2 is considered, with $e_1 = y_2z_1$, $d_2 = w_2x_2$ and $e_2 = y_2w_2$. In **Step5a** the vertex z_2 is located inside the triangle $\Delta w_2z_1y_2$, while in **Step5b** the vertex z_2 is located inside the triangle $\Delta z_1w_2y_2$. It is proven that for an edge $d_4 = w_4x_4$ that deletes y_2z_2 one of the edges y_2w_4 or w_4z_2 is intersecting u_1v_1.

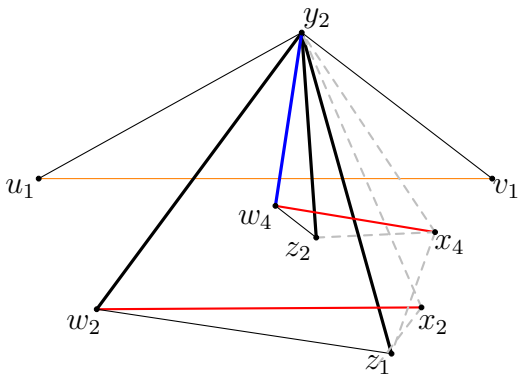
Step5a: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup Z1$ entail:

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge & \mathsf{inside}(w_2, z_1, y_2, z_2) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_2) & (e_3 = y_2z_2) \\
\wedge & \mathsf{deleting}(y_2, z_2, w_4, x_4) & (d_4 = w_4x_4) \\
\wedge\neg & \mathsf{deleting}(y_2, z_1, x_4, w_4) & (\leftarrow orderF) \\
\wedge\neg & \mathsf{deleting}(y_2, z_2, x_4, w_4) & (\leftarrow Step3) \\
\rightarrow & \mathsf{intersection}(u_1, v_1, y_2, w_4) & (d_3(Z), e_4 = y_2w_4(Step2/4/6)) \\
\vee & \mathsf{intersection}(u_1, v_1, w_4, z_2) & (d_3(Y), e_4 = w_4z_2(Step4/8/9))
\end{array}
$$

The edge e_4 is either y_2w_4 or w_4z_2 and e_4 is intersecting u_1v_1.



(**a**) The edge w_4z_2 is intersecting with u_1v_1



(**b**) The edge y_2w_4 is intersecting with u_1v_1

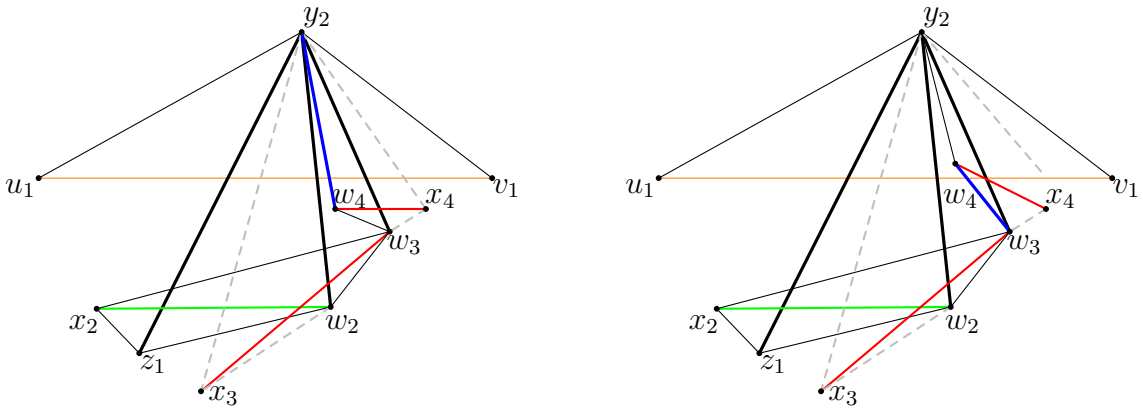
Figure 42: Illustration of two possible cases in Step5a

Step6: (see Fig. 43) Next, the case where the edges $d_2 = w_2x_2$ and $d_3 = w_3x_3$ are both of type Z and an edge $d_4 = w_4x_4$ deleting y_2w_3 is considered. It is proven that one of the edges y_2w_4 or w_4w_3 resulting from the intersection of d_4 and e_3 intersects u_1v_1.

Step6: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup Z1$ entail:

$$
\begin{array}{lr}
\mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge\ \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge\ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3(Z) = w_3x_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2w_3) \\
\wedge\ \mathsf{deleting}(y_2, w_3, w_4, x_4) & (d_4 = w_4x_4) \\
\wedge\neg\mathsf{deleting}(y_2, z_1, x_3, w_3) & (\leftarrow orderF) \\
\wedge\neg\mathsf{deleting}(y_2, z_1, x_4, w_4) & (\leftarrow orderF) \\
\wedge\neg\mathsf{deleting}(y_2, w_3, x_4, w_4) & (\leftarrow Step3) \\
\rightarrow\ \mathsf{intersection}(u_1, v_1, y_2, w_4) & (d_4(Z), e_4 = y_2w_4(\rightarrow Step2/4/6)) \\
\vee\ \mathsf{intersection}(u_1, v_1, w_4, w_3) & (d_4(Y), e_4 = w_4w_3(\rightarrow Step4/8/9) \\
\vee\ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Part2, Step2))
\end{array}
$$

The case that also x_3w_3 deletes y_2w_2 is considered in Part2,Step2, where it is proven that the sequence is continued in another way in this case. Therefore, the edge e_4 is either y_2w_4 or w_4w_3 and e_4 is intersecting u_1v_1.



(**a**) The edge y_2w_4 is intersecting with u_1v_1 (**b**) The edge w_3w_4 is intersecting with u_1v_1

Figure 43: Illustration of two possible cases in Step6

Step7: Now the case that for the edge $e_4 = y_2z_2$, the vertex z_2 is located inside a triangle formed by the vertices of two previous edges y_2w_2 and y_2w_3 of the sequence is considered. In **Step7a** the vertex z_2 is located inside the triangle $\Delta y_2w_2w_3$, while in **Step7b** the vertex z_2 is located inside the triangle $\Delta y_2w_3w_2$. It is proven that for the edge $d_5 = w_5x_5$ deleting e_4, one of the edges y_2w_5 or w_5z_2 resulting from the intersection of d_5 and e_4 is intersecting u_1v_1.

Step7a: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup Z1$ entail:

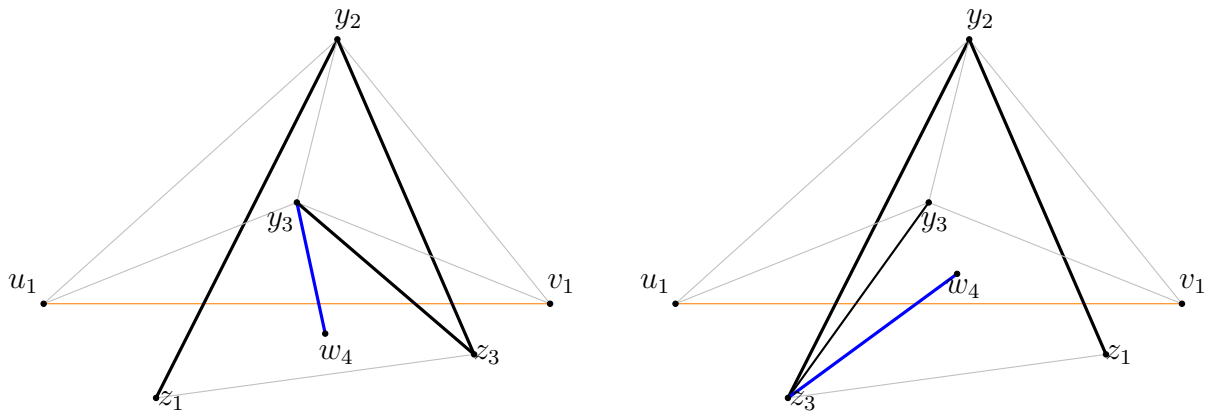
$$
\begin{array}{lr}
\mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge\ \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge\ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3(Z) = w_3x_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2w_3) \\
\wedge\ \mathsf{inside}(y_2, w_2, w_3, z_2) & \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, z_2) & (e_4 = y_2z_2) \\
\wedge\ \mathsf{deleting}(y_2, z_2, w_5, x_5) & (d_5 = w_5x_5) \\
\wedge\neg\mathsf{deleting}(y_2, z_1, x_3, w_3) & (\leftarrow orderF) \\
\wedge\neg\mathsf{deleting}(y_2, z_1, x_4, w_4) & (\leftarrow orderF) \\
\wedge\neg\mathsf{deleting}(y_2, z_2, x_5, w_5) & (\leftarrow Step3) \\
\rightarrow\ \mathsf{intersection}(u_1, v_1, y_2, w_5) & (d_5(Z), e_5 = y_2w_5(\rightarrow Step2/4/6)) \\
\vee\ \mathsf{intersection}(u_1, v_1, w_5, z_2) & (d_5(Y), e_5 = w_5z_2(\rightarrow Step4/8/9) \\
\vee\ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Part2, Step2))
\end{array}
$$

The case that also x_3w_3 deletes y_2w_2 is considered in Part2, Step2, where it is proven that the sequence is continued in another way in this case. Therefore, the edge e_5 is either y_2w_5 or w_5z_2 and e_5 is intersecting u_1v_1.

In Part2 it is proven that every sequence of eligible edges $y_2z_{2_s}$ starting from the switching edge $e_1 = y_2z_1$ can be reduced to one of the cases considered in Step2-7. Therefore, it can be assumed that at least two switching edges y_2z_1 and y_2z_3 exist.

In the following steps the case that the vertex y_3 is located inside a triangle formed by z_1, y_2 and z_3 is considered.

Step8: (see Fig. 44) In this step a vertex w_4 is located inside the same triangle formed by z_1, y_2 and z_3 as y_3. In **Step8a** the vertices y_3 and w_4 are located inside the triangle $\Delta z_3 z_1 y_2$, while in **Step8b** the vertices y_3 and w_4 are located inside the triangle $\Delta z_1 z_3 y_2$. In **Step8a1** resp. **Step8b1** it is proven that one of the edges $w_4 z_3$ or $y_3 w_4$ is intersecting $u_1 v_1$. In the remaining test of **Step8a** and **Step8b** it is proven that u_1 is not located inside a triangle formed by y_2, z_3 and w_4 resp. y_3, z_3 and w_4.



(**a**) The edge $y_3 w_4$ is intersecting with $u_1 v_1$ (**b**) The edge $w_4 z_3$ is intersecting with $u_1 v_1$

Figure 44: Two possibilities to continue the sequence of eligible edges if y_3 and w_4 are located inside a triangle formed by z_1, y_2 and z_3

Step8a1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$
\begin{aligned}
& \mathsf{intersection}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = y_2 z_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge \neg & \mathsf{inside}(z_3, z_1, y_2, u_1) && (\leftarrow Step1 - 7) \\
\wedge \neg & \mathsf{inside}(z_3, z_1, y_2, v_1) && (\leftarrow Step1 - 7) \\
\wedge\ & \mathsf{inside}(z_3, z_1, y_2, y_3) && \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_3, z_3) && (e_3 = y_3 z_3) \\
\wedge\ & \mathsf{inside}(z_3, z_1, y_2, w_4) && \\
\rightarrow\ & \mathsf{intersection}(u_1, v_1, w_4, z_3) && (e_4 = w_4 z_3) \\
\vee\ & \mathsf{intersection}(u_1, v_1, y_3, w_4) && (e_4 = y_3 w_4)
\end{aligned}
$$

The edge e_4 is either $w_4 z_3$ or $y_3 w_4$ and e_4 is intersecting $u_1 v_1$.

Step9: In **Step9a** the vertex y_3 is located inside the triangle $\Delta z_3 z_1 y_2$, while in **Step9b** the vertex y_3 is located inside the triangle $\Delta z_1 z_3 y_2$. Furthermore, the edge $y_3 w_4$ is intersecting $z_1 z_3$. In **Step9a1** resp. **Step9b1** it is proven that the edge $y_3 w_4$ is also intersecting $u_1 v_1$. In the remaining test of **Step9a** and **Step9b** it is proven that u_1 is not located inside a triangle formed by y_3, z_3 and w_4.

Step9a1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entail:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1 v_1, e_1 = y_2 z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1-7) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1-7) \\
\wedge \ \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3 z_3) \\
\wedge \ \mathsf{intersection}(z_1, z_3, y_3, w_4) & \\
\rightarrow \ \mathsf{intersection}(u_1, v_1, y_3, w_4) & (e_4 = y_3 w_4)
\end{array}
$$

The edge e_4 is $y_3 w_4$ and e_4 is intersecting $u_1 v_1$.

The proofs of Step8 and Step9 can be adapted by changing the labeling of the vertices accordingly, for any eligible edges of the sequence, where an eligible vertex is located inside a triangle formed by the vertices of two previous switching edges, that have a common vertex.

Proving the intersection of the eligible edges with $u_1 v_1$ inductively

In Step1 it was proven that all edges $y_{1_n} z_1$ intersect $u_1 v_1$. Therefore, also the switching edge $y_2 z_1$ intersects $u_1 v_1$. In Step2-7 it was proven that all edges $y_2 z_{2_s}$ intersect $u_1 v_1$. Therefore, also the switching edge $y_2 z_3$ intersects $u_1 v_1$. Furthermore, from the Steps2-7 it follows that for an edge $d_i = w_i x_i$ deleting the switching edge $y_2 z_3$, the edge $y_{3_1} z_3$ with $y_{3_1} = w_i$ is intersecting with $u_1 v_1$. In Part3 Steps1-3 (Section 7.2.3) it is proven that for all $n \geq 2$ the vertex y_{3_n} is located inside a triangle formed by z_1, y_2 and z_3 or inside a triangle formed by y_2, z_3 and y_{3_1}. From Part1 Step4 it follows that the edge $y_{3_n} z_3$ is intersecting with $u_1 v_1$ and consequently also the switching edge $y_4 z_3$ is intersecting $u_1 v_1$.

In the remaining steps of Part3 it is proven that for all $k \geq 2$ and all s one of the following conditions is satisfied:

- (a) the vertex z_{2k_s} is located inside a triangle formed by y_{2k-2}, z_{2k-1} and y_{2k}
- (b) the vertex z_{2k_s} is located inside the same triangle formed by z_{2k-3}, y_{2k-2} and z_{2k-1} as y_{2k}
- (c) the edges $y_{2k} z_{2k_s}$ and $z_{2k-3} z_{2k-1}$ are intersecting.

An edge $y_{2k}z_{2k_s}$ satisfying one of the conditions (a)-(c) intersects with u_1v_1 according to Part1, Step4 (condition (a)), Part1, Step8 (condition (b)) or Part1, Step9 (condition (c)). Consequently one of the conditions (a)-(c) also holds for the switching vertex z_{2k+1} resp. the switching edge $y_{2k}z_{2k+1}$. Analogously for all $k \geq 2$ and all n the vertex y_{2k+1_n} is located inside a triangle formed by z_{2k-1}, y_{2k} and z_{2k+1}, inside the same triangle formed by y_{2k-1}, z_{2k-1} and y_{2k} as z_{2k+1} or the edges $y_{2k+1_n}z_{2k+1}$ and $y_{2k-2}y_{2k}$ are intersecting. Therefore, one of these conditions also holds for the vertex y_{2k+2} resp. the switching edge $y_{2k+2}z_{2k+1}$.

Therefore, all eligible edges are intersecting with u_1v_1 and since the graph is assumed to be finite the sequence has to contain a redundancy cycle or an edge $y_{f(t)}z_{g(t)}$ which is not deleted by another edge. If the edge $y_{f(t)}z_{g(t)}$ is not deleted by another edge, then $y_{f(t)}z_{g(t)}$ is an edge in F intersecting u_1v_1.

7.2.2 Proof of Part2: No Redundancy Cycle Formed by Incident Edges

In this part of the proof it is proven that no redundancy cycle can occur among eligible edges, that are incident, i.e all edges have one vertex in common. This part of the proof is divided into the Steps1-6. An overview of the proof scheme is given in Fig. 45. In order to simplify the notation in the flow chart, the notation $\overline{\mathsf{inside}}(u, v, w, x)$ is introduced, which means that x is, depending on the orientation of the triangle, inside the triangle Δuvw or inside the triangle Δvuw, i.e

$$\overline{\mathsf{inside}}(u, v, w, x) \leftrightarrow \mathsf{inside}(u, v, w, x) \vee \mathsf{inside}(v, u, w, x)$$

Furthermore, $\overline{\mathsf{inside*}}(u, v, w, x/^*y)$ means that x or y is located inside the triangle Δuvw or inside the triangle Δvuw, i.e

$$\begin{aligned}&\overline{\mathsf{inside*}}(u, v, w, x/^*y)\\ \leftrightarrow&\mathsf{inside}(u, v, w, x) \vee \mathsf{inside}(v, u, w, x) \vee \mathsf{inside}(u, v, w, y) \vee \mathsf{inside}(v, u, w, y)\end{aligned}$$

It is proven first in the Steps1-5 that any redundancy cycle among the eligible edges of length greater than 2, can be reduced to a redundancy cycle of length 2. Then in Step6 for a redundancy cycle of length 2, it is proven that the deleting edges can be chosen in an alternative way, such that the redundancy cycle is avoided. The proof is explicitly given for a sequence starting with e_1, which is the first edge that occurs twice, but can be considered for any other eligible edges, by adapting the labeling accordingly. The tests of all the following proof tasks of Part2 can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/part2.

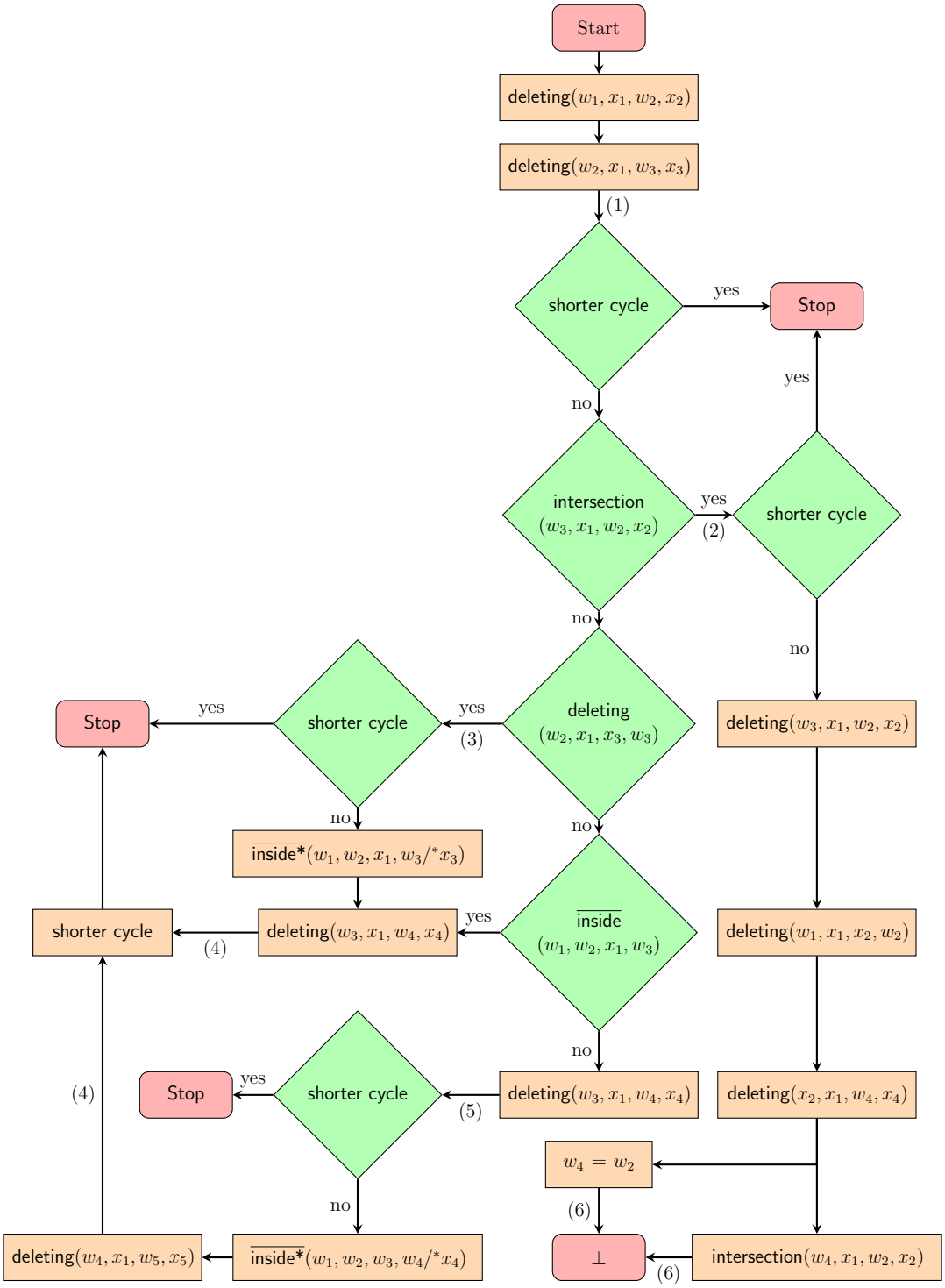
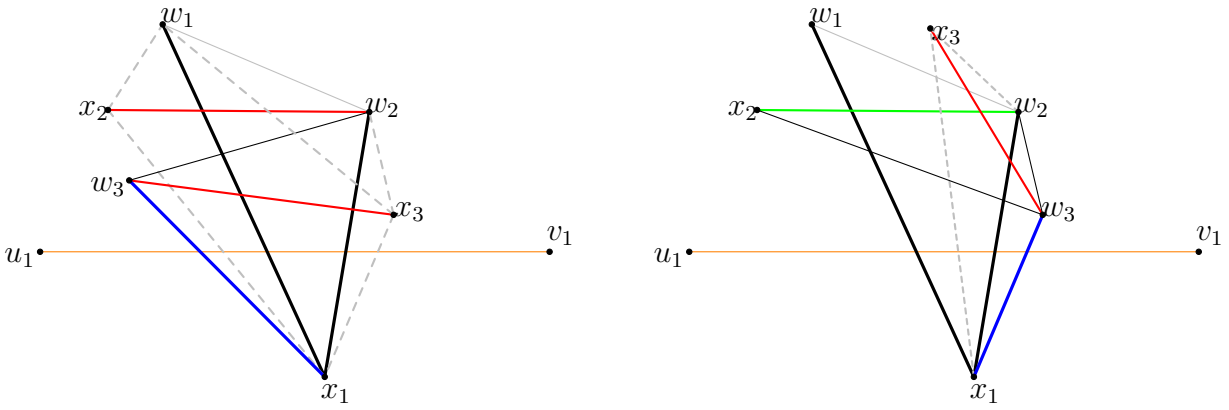


Figure 45: The proof scheme of Part2, the numbers labeling the arrows correspond to the proof steps.

Step1: (see Fig. 46) In this step an edge w_1x_1 is deleted by an edge w_2x_2 and w_2x_1 is deleted by w_3x_3. This implies that there exits an edge d_2' deleting w_1x_1, leading to the edge $e_2 = w_3x_1$, the edges w_2x_2 and w_3x_3 are intersecting, one of the edges w_2x_2 or w_3x_3 is in F or the vertex w_4 is located inside a triangle formed by w_1, x_1 and w_2.



(**a**) The edge w_3x_3 is also deleting w_1x_1

(**b**) The edge w_2x_2 is in F

Figure 46: Two possibilities to continue the sequence of eligible after Step1

It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the following formula (the atoms displayed in light gray are commented out in the tests, because they are not needed for proving the conclusion):

$$
\begin{array}{lll}
 & \mathsf{intersection}(u_1, v_1, w_1, x_1) & (e_0 = u_1v_1, e_1 = w_1x_1) \\
\wedge & \mathsf{deleting}(w_1, x_1, w_2, x_2) & (d_2 = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_2, x_1) & (e_2 = w_2x_1) \\
\wedge & \mathsf{deleting}(w_2, x_1, w_3, x_3) & (d_3 = w_3x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, x_1) & (e_3 = w_3x_1) \\
\rightarrow & \mathsf{deleting}(w_1, x_1, w_3, x_2) & (d_2' = w_3x_2) \\
\vee & \mathsf{deleting}(w_1, x_1, w_3, x_3) & (d_2' = w_3x_3) \\
\vee & \mathsf{intersection}(w_3, x_1, w_2, x_2) & (\rightarrow \mathit{Step2}) \\
\vee & \mathsf{deleting}(w_2, x_1, x_3, w_3) & (d_{3b} = x_3w_3 (\rightarrow \mathit{Step3})) \\
\vee & \mathsf{inside}(w_1, x_1, w_2, w_3) & (\rightarrow \mathit{Step4}) \\
\vee & \mathsf{inside}(w_1, w_2, x_1, w_3) & (\rightarrow \mathit{Step4}) \\
\vee & \mathsf{F}(w_2, x_2) & (\rightarrow \mathit{Step5})
\end{array}
$$

In the following proof steps the cases resulting from the implication in Step1 are considered. For all of the following proof steps it is assumed that the edge w_1x_1 is deleted by the edge w_2x_2.

Step2: In this step the edge w_2x_1 is deleted by an edge w_3x_3 and w_3x_1 intersects w_2x_2. This implies that there exists another edge d_2' deleting w_1x_1 leading to $e_2' = w_3x_1$, but e_2' is not deleted by w_2x_2 or a redundancy cycle of length two containing the edges $e_2 = w_2x_1$ and $e_3 = w_3x_1$ exists. In **Step2a** w_2x_2 is only intersecting w_3x_1, in **Step2b** w_3x_3 is also deleting w_2x_1 and in **Step2c** w_2x_2 is also deleting w_3x_1.

Step2a: First an intersection between w_3x_1 and w_2x_2 is considered.

It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

$$
\begin{array}{lll}
 & \mathsf{intersection}(u_1, v_1, w_1, x_1) & (e_0 = u_1v_1, e_1 = w_1x_1) \\
\wedge & \mathsf{deleting}(w_1, x_1, w_2, x_2) & (e_1 = w_1x_1, d_2 = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_2, x_1) & (e_2 = w_2x_1) \\
\wedge & \mathsf{deleting}(w_2, x_1, w_3, x_3) & (d_3 = w_3x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, x_1) & (e_3 = w_3x_1) \\
\wedge & \mathsf{intersection}(w_3, x_1, w_2, x_2) & \\
\rightarrow & \mathsf{deleting}(w_1, x_1, w_3, x_2) & (d_2' = w_3x_2) \\
\vee & \mathsf{deleting}(w_1, x_1, w_3, x_3) & (d_2' = w_3x_3) \\
\vee & \mathsf{deleting}(w_2, x_1, x_3, w_3) & (\rightarrow \mathit{Step2b}) \\
\vee & \mathsf{deleting}(w_3, x_1, w_2, x_2) & (d_4 = w_2x_2(\rightarrow \mathit{Step2c}))
\end{array}
$$

The case that also x_3w_3 is deleting w_2x_1 is considered in Step2b, while the case that w_2x_2 is not only intersecting, but also deleting w_3x_1 is considered in Step2c. For the remaining two cases an alternative deleting edge d_2' can be chosen leading to $e_2 = w_3x_1$.

Step2b: Next the case that also x_3w_3 is deleting w_2x_1 is considered.

It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

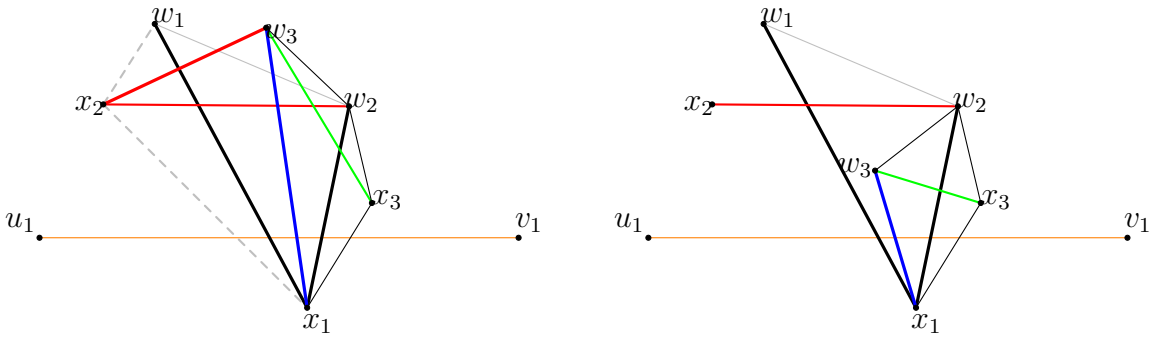
$$
\begin{array}{lll}
 & \mathsf{intersection}(u_1, v_1, w_1, x_1) & (e_0 = u_1v_1, e_1 = w_1x_1) \\
\wedge & \mathsf{deleting}(w_1, x_1, w_2, x_2) & (e_1 = w_1x_1, d_2 = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_2, x_1) & (e_2 = w_2x_1) \\
\wedge & \mathsf{deleting}(w_2, x_1, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, x_1) & (d_3 = d_{3a}, e_3 = w_3x_1) \\
\wedge & \mathsf{deleting}(w_2, x_1, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, x_3, x_1) & (d_3 = d_{3b}, e_3 = x_3x_1) \\
\wedge & \mathsf{intersection}(w_3, x_1, w_2, x_2) & (d_3 = d_{3a}, d_4 = d_2 = w_2x_2) \\
\rightarrow & \mathsf{deleting}(w_1, x_1, w_3, x_2) & (d'_2 = w_3x_2) \\
\vee & \mathsf{deleting}(w_1, x_1, w_3, x_3) & (d'_2 = w_3x_3) \\
\vee & \mathsf{deleting}(w_1, x_1, x_3, x_2) & (d'_2 = x_3x_2) \\
\vee & \mathsf{deleting}(w_1, x_1, x_3, w_3) & (d'_2 = x_3w_3)
\end{array}
$$

Step2c: Now the case that w_3x_1 is deleted by w_2x_2 is considered. For this, it is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, w_1, x_1) & (e_0 = u_1v_1, e_1 = d_1 = w_1x_1) \\
\wedge & \mathsf{deleting}(w_1, x_1, w_2, x_2) & (e_1 = w_1x_1, d_2 = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_2, x_1) & (e_2 = w_2x_1) \\
\wedge & \mathsf{deleting}(w_2, x_1, w_3, x_3) & (d_3 = w_3x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, x_1) & (e_3 = w_3x_1) \\
\wedge & \mathsf{deleting}(w_3, x_1, w_2, x_2) & (d_4 = d_2 = w_2x_2) \\
\wedge\neg & \mathsf{deleting}(w_2, x_1, x_3, w_3) & (\leftarrow \mathit{Step2b}) \\
\rightarrow & \mathsf{deleting}(w_1, x_1, x_2, w_2) & (d_{2b} = x_2w_2 (\rightarrow \mathit{Step6})
\end{array}
$$

As a result of Step2, the edges d_2 and d_3 can be chosen in a way such that e_3 and d_2 do not intersect or a redundancy cycle of size 2 occurs. The case of a redundancy cycle of size 2 is considered in Step6.

Step3: (see Fig. 47) Next the case that w_2x_1 is deleted by w_3x_3 and x_3w_3 is considered. The existence of both deleting edges w_3x_3 and x_3w_3 implies, that there exists another edge d_2' deleting w_1x_1 leading to $e_2' = w_3x_1$ or $e_2' = x_3x_1$ or one of the vertices w_3 or x_3 is located inside a triangle formed by w_1, x_1 and w_2.



(**a**) The edge w_3x_2 is also deleting w_1x_1

(**b**) The vertex w_3 is located inside the triangle $\Delta w_1x_1w_2$

Figure 47: Two possibilities to continue the sequence of eligible edges after Step3

It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

	$\mathsf{intersection}(u_1, v_1, w_1, x_1)$	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	$\mathsf{deleting}(w_1, x_1, w_2, x_2)$	$(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_2, x_1)$	$(e_2 = w_2x_1)$
$\wedge$	$\mathsf{deleting}(w_2, x_1, w_3, x_3)$	$(d_{3a} = w_3x_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, x_1)$	$(d_3 = d_{3a}, e_3 = w_3x_1)$
$\wedge$	$\mathsf{deleting}(w_2, x_1, x_3, w_3)$	$(d_{3b} = x_3w_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, x_3, x_1)$	$(d_3 = d_{3b}, e_3 = x_3x_1)$
$\rightarrow$	$\mathsf{deleting}(w_1, x_1, w_3, x_2)$	$(d_2' = w_3x_2)$
$\vee$	$\mathsf{deleting}(w_1, x_1, w_3, x_3)$	$(d_2' = w_3x_3)$
$\vee$	$\mathsf{deleting}(w_1, x_1, x_3, w_3)$	$(d_2' = x_3w_3)$
$\vee$	$\mathsf{deleting}(w_1, x_1, x_3, x_2)$	$(d_2' = x_3x_2)$
$\vee$	$\mathsf{inside}(w_1, x_1, w_2, w_3) \vee \mathsf{inside}(w_1, x_1, w_2, x_3)$	$(\rightarrow \mathit{Step4})$
$\vee$	$\mathsf{inside}(x_1, w_1, w_2, w_3) \vee \mathsf{inside}(x_1, w_1, w_2, x_3)$	$(\rightarrow \mathit{Step4})$

If no alternative deleting edge for the edge w_1x_1 exists, the labeling of the vertices of $d_3 = w_3x_3$ is chosen such that w_3 is located inside a triangle formed by w_1, x_1 and x_2 and $e_3 = w_3x_1$ is considered as the next eligible edge.

Step4: In **Step4a** and **Step4b** the case that the vertex w_3 is located inside the triangle $\Delta w_1x_1w_2$ is regarded. The similar case that w_3 is located inside the triangle $\Delta x_1w_1w_2$ is regarded in **Step4c** and **Step4d**. Furthermore, in all cases the edge w_3x_1, is deleted by an edge w_4x_4. In **Step4a1**, **Step4b1**, **Step4c1** and **Step4d1** only the edge w_4x_4 deletes w_3x_1, while in **Step4a2**, **Step4b2**, **Step4c2** and **Step4d2** also the edge x_4w_4 deletes w_3x_1.

It is first proven in **Step4a** resp. **Step4c** that if the edge w_4x_1 is intersecting with the edge w_2x_2, then there exists another edge d_2' deleting w_1x_1 leading to $e_2' = w_4x_1$ or another edge d_3' deleting w_2x_1 leading to $e_3' = w_4x_1$. Furthermore, it is proven in **Step4a3** resp. **Step4c3** that if the edge w_4x_1 is deleted by w_3x_3, then the edge x_3x_4 is also deleting w_1x_1 or w_2x_1. Afterwards it is proven in **Step4b** resp. **Step4d** that there exists another edge d_2' deleting w_1x_1 leading to $e_2' = w_4x_1$ or another edge d_3' deleting w_2x_1 leading to $e_3' = w_4x_1$ or one of the vertices w_4 or x_4 is located inside the triangle $\Delta w_1x_1w_2$ **(Step4b)** resp. $\Delta x_1w_1w_2$ **(Step4d)**.

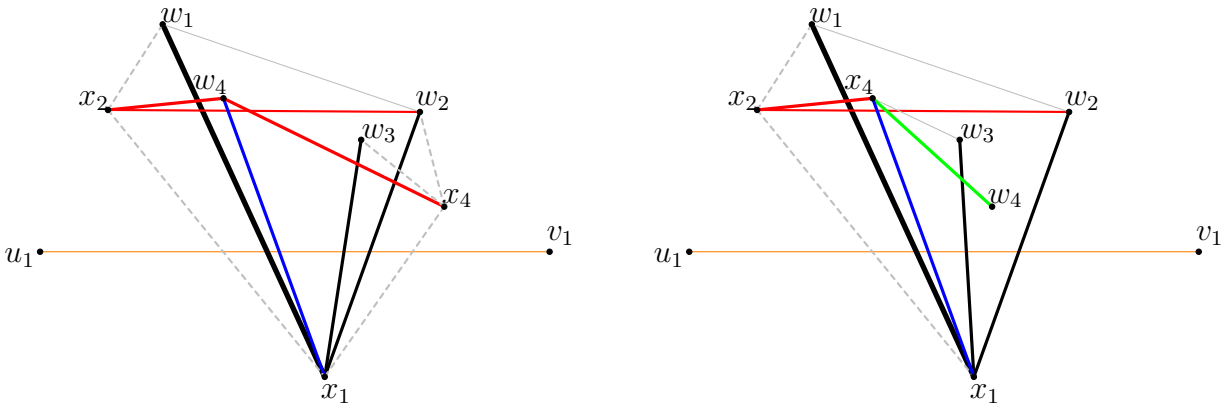
In the following, only the details of the proofs for **Step4a** and **Step4b** are presented. The proofs for **Step4c** and **Step4d** are similar and listed in Appendix 9.4.4.

Step4a1: (see Fig. 48(**a**)) First an intersection between w_3x_1 and w_2x_2 is considered. It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

$$
\begin{array}{lll}
 & \mathsf{intersection}(u_1, v_1, w_1, x_1) & (e_0 = u_1v_1, e_1 = w_1x_1) \\
\wedge & \mathsf{deleting}(w_1, x_1, w_2, x_2) & (d_2 = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_2, x_1) & (e_2 = w_2x_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, x_1) & (e_3 = w_3x_1) \\
\wedge\neg & \mathsf{intersection}(w_3, x_1, w_2, x_2) & (\leftarrow \mathit{Step}3/4) \\
\wedge & \mathsf{inside}(w_1, x_1, w_2, w_3) & \\
\wedge & \mathsf{deleting}(w_3, x_1, w_4, x_4) & (d_4 = w_4x_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_4, x_1) & (e_4 = w_4x_1) \\
\wedge & \mathsf{intersection}(w_4, x_1, w_2, x_2) & \\
\rightarrow & \mathsf{deleting}(w_1, x_1, w_4, x_2) & (d_2' = w_4x_2) \\
\vee & \mathsf{deleting}(w_1, x_1, w_4, x_4) & (d_2' = w_4x_4) \\
\vee & \mathsf{deleting}(w_2, x_1, w_4, x_2) & (d_3' = w_4x_2) \\
\vee & \mathsf{deleting}(w_2, x_1, w_4, x_4) & (d_3' = w_4x_4) \\
\vee & \mathsf{deleting}(w_3, x_1, x_4, w_4) & (d_{4b} = x_4w_4(\mathit{Step}4a2))
\end{array}
$$

Step4a2: (see Fig. 48(**b**)) Now the case that also x_4w_4 is deleting w_3x_1 and x_4x_1 is intersecting with w_2x_2 is considered. It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

$$
\begin{aligned}
& \mathsf{intersection}(u_1, v_1, w_1, x_1) && (e_0 = u_1v_1, e_1 = w_1x_1)\\
\wedge\ & \mathsf{deleting}(w_1, x_1, w_2, x_2) && (d_2 = w_2x_2)\\
\wedge\ & \mathsf{intersection}(u_1, v_1, w_2, x_1) && (e_2 = w_2x_1)\\
\wedge\ & \mathsf{intersection}(u_1, v_1, w_3, x_1) && (e_3 = w_3x_1)\\
\wedge\neg\ & \mathsf{intersection}(w_3, x_1, w_2, x_2) && (\leftarrow \mathit{Step}3/4)\\
\wedge\ & \mathsf{inside}(w_1, x_1, w_2, w_3)\\
\wedge\ & \mathsf{deleting}(w_3, x_1, w_4, x_4) && (d_{4a} = w_4x_4)\\
\wedge\ & \mathsf{intersection}(u_1, v_1, w_4, x_1) && (d_4 = d_{4a}, e_4 = w_4x_1)\\
\wedge\ & \mathsf{intersection}(w_4, x_1, w_2, x_2)\\
\wedge\ & \mathsf{deleting}(w_3, x_1, x_4, w_4) && (d_{4b} = x_4w_4)\\
\wedge\ & \mathsf{intersection}(u_1, v_1, x_4, x_1) && (d_4 = d_{4b}, e_4 = x_4x_1)\\
\wedge\ & \mathsf{intersection}(x_4, x_1, w_2, x_2)\\
\rightarrow\ & \bot
\end{aligned}
$$



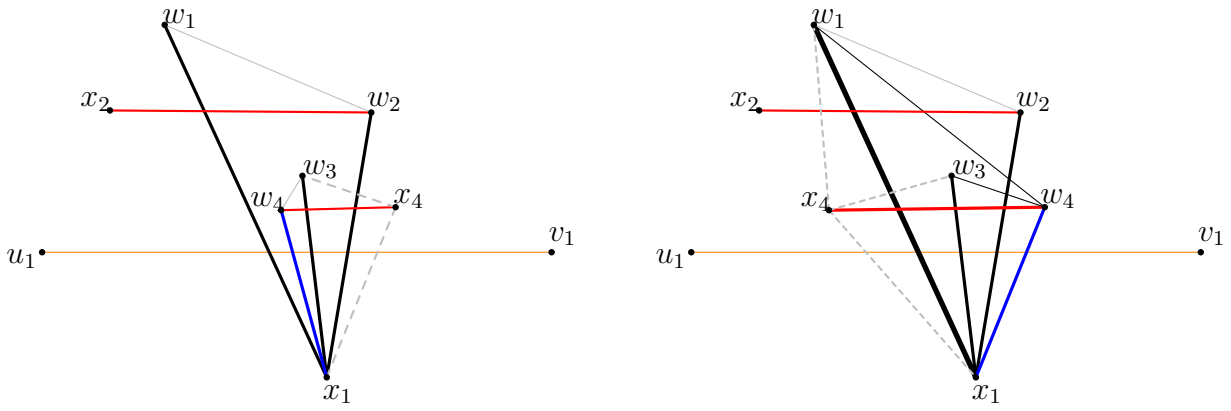
(**a**) The edge w_4x_2 is also deleting w_1x_1 (**b**) The edge x_4x_2 is also deleting w_1x_1

Figure 48: Two possibilities to continue the sequence of eligible edges after Step4

The proof of Step4a3 is similar and is listed in Appendix 9.4.4.

As a result of the Step 4a it can be assumed that for all vertices y_{1_s} located inside that triangle $\Delta w_1x_1w_2$, the eligible edge $y_{1_s}x_1$ is not intersecting with w_2x_2.

Step4b1: Next the question whether w_4 is located inside the triangle $\Delta w_1x_1w_2$ and the cases that can occur if w_4 is not located inside the triangle $\Delta w_1x_1w_2$ are considered (see Fig. 49) .



(**a**) w_4 is located inside the triangle $\Delta w_1x_1w_2$ (**b**) The edge w_4x_4 is also deleting w_1x_1

Figure 49: Two possibilities to continue the sequence of eligible edges after Step4

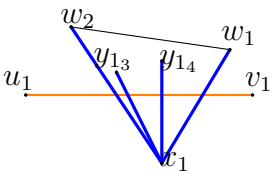
It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

	$\mathsf{intersection}(u_1, v_1, w_1, x_1)$	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	$\mathsf{deleting}(w_1, x_1, w_2, x_2)$	$(d_2 = w_2x_2)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_2, x_1)$	$(e_2 = w_2x_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, x_1)$	$(e_3 = w_3x_1)$
$\wedge\neg$	$\mathsf{intersection}(w_3, x_1, w_2, x_2)$	$(\leftarrow Step3/4)$
$\wedge$	$\mathsf{inside}(w_1, x_1, w_2, w_3)$	
$\wedge$	$\mathsf{deleting}(w_3, x_1, w_4, x_4)$	$(d_4 = w_4x_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_4, x_1)$	$(e_4 = w_4x_1)$
$\rightarrow$	$\mathsf{deleting}(w_1, x_1, w_4, x_2)$	$(d'_2 = w_4x_2)$
$\vee$	$\mathsf{deleting}(w_1, x_1, w_4, x_4)$	$(d'_2 = w_4x_4)$
$\vee$	$\mathsf{deleting}(w_2, x_1, w_4, x_2)$	$(d'_3 = w_4x_2)$
$\vee$	$\mathsf{deleting}(w_2, x_1, w_4, x_4)$	$(d'_3 = w_4x_4)$
$\vee$	$\mathsf{deleting}(w_3, x_1, x_4, w_4)$	$(d_{4b} = x_4w_4(\rightarrow Step4b2))$
$\vee$	$\mathsf{inside}(w_1, x_1, w_2, w_4)$	$(w_4\ as\ w_3\ in\ Step4)$

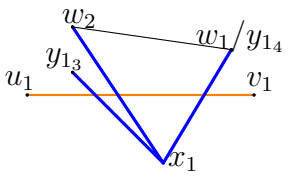
Step4b2: Now the case that also x_4w_4 is deleting w_3x_1 is considered. It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

$$
\begin{array}{lll}
 & \mathsf{intersection}(u_1, v_1, w_1, x_1) & (e_0 = u_1v_1, e_1 = w_1x_1) \\
\wedge & \mathsf{deleting}(w_1, x_1, w_2, x_2) & (d_2 = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_2, x_1) & (e_2 = w_2x_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, x_1) & (e_3 = w_3x_1) \\
\wedge\neg & \mathsf{intersection}(w_3, x_1, w_2, x_2) & (\leftarrow \mathit{Step}3/4) \\
\wedge & \mathsf{inside}(w_1, x_1, w_2, w_3) & \\
\wedge & \mathsf{deleting}(w_3, x_1, w_4, x_4) & (d_{4a} = w_4x_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_4, x_1) & (d_4 = d_{4a}, e_4 = w_4x_1) \\
\wedge & \mathsf{deleting}(w_3, x_1, x_4, w_4) & (d_{4b} = x_4w_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, x_4, x_1) & (d_4 = d_{4b}, e_4 = x_4x_1) \\
\rightarrow & \mathsf{deleting}(w_1, x_1, w_4, x_4) & (d'_2 = w_4x_4) \\
\vee & \mathsf{deleting}(w_1, x_1, x_4, w_4) & (d'_2 = x_4w_4) \\
\vee & \mathsf{deleting}(w_2, x_1, w_4, x_4) & (d'_3 = w_4x_4) \\
\vee & \mathsf{deleting}(w_2, x_1, x_4, w_4) & (d'_3 = x_4w_4) \\
\vee & \mathsf{inside}(w_1, x_1, w_2, w_4) & (w_4 \ as \ w_3 \ in \ \mathit{Step}4) \\
\vee & \mathsf{inside}(w_1, x_1, w_2, x_4) & (x_4 \ as \ w_3 \ in \ \mathit{Step}4)
\end{array}
$$

As a result of Step4 an eligible edge $y_{1_s}x_1$, where y_{1_s} is located inside the triangle $\Delta w_1x_1w_2$ resp. $\Delta x_1w_1w_2$ cannot be part of a shortest redundancy cycle, that also contains w_1x_1 (see Fig. 50). This can be assumed because the border vertex w_1 is not located inside the triangle $\Delta w_1x_1w_2$ resp. $\Delta x_1w_1w_2$ and therefore a redundancy cycle containing the edge w_1x_1 cannot occur if for all following eligible edges $y_{1_{s'}}x_1$, with $s' > s$ the vertex $y_{1_{s'}}$ is located inside the triangle $\Delta w_1x_1w_2$ resp. $\Delta x_1w_1w_2$.



(**a**) A shortest cyclic sequence cannot contain the edge w_1x_1



(**b**) A possible shortest cyclic sequence containing w_1x_1

Figure 50: For y_{1_3} located inside $\Delta x_1w_1w_2$, w_1x_1 is not part of a shortest redundancy cycle.

Step5: As in the previous steps an edge w_1x_1 is deleted by an edge w_2x_2 and w_2x_1 is deleted by w_3x_3. In addition the edge w_3x_1 is deleted by an edge w_4x_4. It is proven that then there exists another edge d'_3 deleting w_2x_1 leading to $e'_3 = w_4x_1$, w_4x_1 is intersecting w_3x_3, also the edge x_4w_4 is deleting w_3x_1 or w_4 or x_4 is located inside a triangle formed by x_1, w_2 and w_3.

It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

	$\mathsf{intersection}(u_1, v_1, w_1, x_1)$	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	$\mathsf{deleting}(w_1, x_1, w_2, x_2)$	$(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_2, x_1)$	$(e_2 = w_2x_1)$
$\wedge$	$\mathsf{deleting}(w_2, x_1, w_3, x_3)$	$(d_3 = w_3x_3)$
$\wedge\neg$	$\mathsf{deleting}(w_1, x_1, w_3, x_2)$	$(shortest\ cycle)$
$\wedge\neg$	$\mathsf{deleting}(w_1, x_1, w_3, x_3)$	$(shortest\ cycle)$
$\wedge\neg$	$\mathsf{deleting}(w_1, x_1, x_3, w_3)$	$(\leftarrow orderF)$
$\wedge\neg$	$\mathsf{deleting}(w_2, x_1, x_3, w_3)$	$(\leftarrow Step2)$
$\wedge\neg$	$\mathsf{intersection}(w_3, x_1, w_2, x_2)$	$(\leftarrow Step3)$
$\wedge\neg$	$\mathsf{inside}(w_1, x_1, w_2, w_3)$	$(\leftarrow Step4)$
$\wedge\neg$	$\mathsf{inside}(x_1, w_1, w_2, w_3)$	$(\leftarrow Step4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, x_1)$	$(e_3 = w_3x_1)$
$\wedge$	$\mathsf{deleting}(w_3, x_1, w_4, x_4)$	$(d_4 = w_4x_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_4, x_1)$	$(e_4 = w_4x_1)$
$\rightarrow$	$\mathsf{deleting}(w_2, x_1, w_4, x_3)$	$(d'_3 = w_4x_3)$
$\vee$	$\mathsf{deleting}(w_2, x_1, w_4, x_4)$	$(d'_3 = w_4x_4)$
$\vee$	$\mathsf{intersection}(w_4, x_1, w_3, x_3)$	$(\rightarrow Step2)$
$\vee$	$\mathsf{deleting}(w_3, x_1, x_4, w_4)$	$(d_{4b} = x_4w_4(\rightarrow Step3))$
$\vee$	$\mathsf{inside}(x_1, w_2, w_3, w_4)$	$(\rightarrow Step4)$
$\vee$	$\mathsf{inside}(x_1, w_3, w_2, w_4)$	$(\rightarrow Step4)$

As a result of Step5 a shorter cycle exists or a case occurs that corresponds to one of the cases considered in Steps2-4 with an adapted labeling. Therefore, for all further edges of the sequence it can be assumed that the vertex y_{1_s} is located inside one of the triangles $\Delta x_1w_2w_3$ or $\Delta x_1w_3w_2$ if no shorter cycle exists.

Step6: (see Fig. 51) Finally the case of a redundancy cycle of size 2 is considered. According to Step2 it can be assumed that this cycle contains the edges w_3x_1 and w_2x_1, where w_2x_2 deletes w_3x_1 and w_3x_3 deletes w_2x_1. From the implication proved in Step2c it follows that also x_2w_2 is deleting w_1x_1. Therefore, in the sequence of deleting and eligible edges d_2 can be chosen as x_2w_2, leading to $e_2 = x_2x_1$. It remains to consider if also the choice of $d_2 = x_2w_2$ can lead to a redundancy cycle containing the edge w_2x_1 or an edge intersecting with d_2, such that the edge d_2 would be contained in the sequence twice. According to the previous proof steps such a redundancy cycle could be reduced to a length of 2 or the edge deleting x_2x_1 is in F. In case of a cycle of length 2 an edge w_4x_4 deleting x_2x_1, such that w_4x_1 is intersecting the edge w_2x_2 **(Step6a1)** or $w_4 = w_2$ **(Step6a2)** has to exist.

Step6a1: (see Fig. 51(**a**)) First the case that w_4x_4 is deleting x_2w_1 and w_4x_1 intersecting the edge w_2x_2 is considered. It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraph} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion):

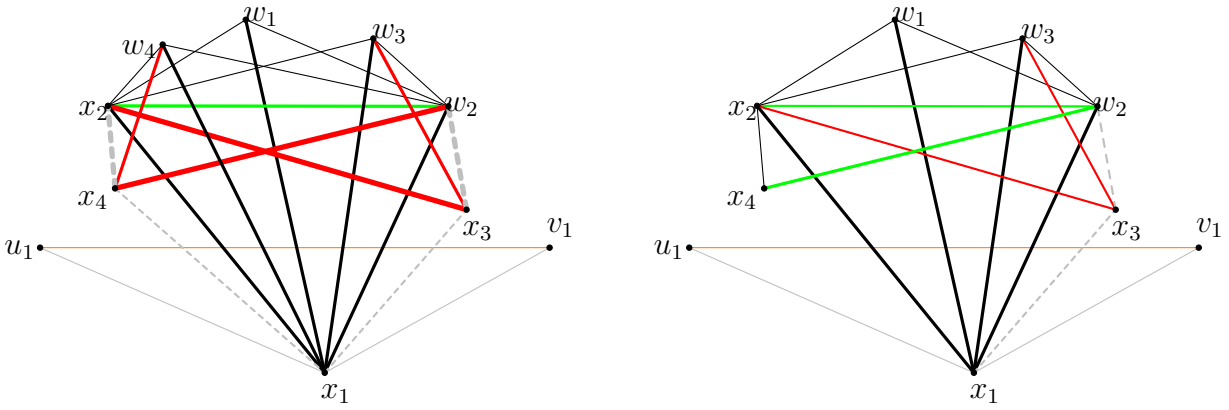
$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, w_1, x_1) & (e_0 = u_1v_1, e_1 = w_1x_1) \\
\wedge\ \mathsf{deleting}(w_1, x_1, w_2, x_2) & (d_{2a} = w_2x_2) \\
\wedge\ \mathsf{intersection}(u_1, v_1, w_2, x_1) & (d_2 = d_{2a}, e_2 = w_2x_1) \\
\wedge\ \mathsf{deleting}(w_2, x_1, w_3, x_3) & (d_2 = d_{2a}, d_3 = w_3x_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, w_3, x_1) & (d_2 = d_{2a}, e_3 = w_3x_1) \\
\wedge\ \mathsf{deleting}(w_3, x_1, w_2, x_2) & (d_2 = d_{2a}, d_4 = w_2x_2) \\
\wedge\ \mathsf{deleting}(w_1, x_1, x_2, w_2) & (d_{2b} = x_2w_2) \\
\wedge\ \mathsf{intersection}(u_1, v_1, x_2, x_1) & (d_2 = d_{2b}, e_2 = x_2x_1) \\
\wedge\ \mathsf{deleting}(x_2, x_1, w_4, x_4) & (d_2 = d_{2b}, d_3 = w_4x_4) \\
\wedge\ \mathsf{intersection}(u_1, v_1, w_4, x_1) & (d_2 = d_{2b}, e_3 = w_4x_1) \\
\wedge\ \mathsf{intersection}(w_4, x_1, w_2, x_2) & \\
\rightarrow\ \bot &
\end{array}
$$

The contradiction is obtained for all edges w_4x_4 deleting x_2x_1, where w_4x_1 is intersecting with w_2x_2. Therefore, also if $w_4 \neq w_1$ a redundancy cycle of length 2 formed by the edges x_2x_1 and w_4x_1 is not possible.

Step6a2: (see Fig. 51(**b**)) It remains the case that the edge w_2x_4 is deleting the edge x_2x_1. For this it is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraph} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails the formula below (some of the atoms are displayed in light gray and commented out in the tests, because they are not needed for proving the conclusion)::

$$
\begin{array}{lr}
\mathsf{intersection}(u_1, v_1, w_1, x_1) & (e_0 = u_1v_1, e_1 = w_1x_1) \\
\wedge\ \mathsf{deleting}(w_1, x_1, w_2, x_2) & (d_{2a} = w_2x_2) \\
\wedge\ \mathsf{intersection}(u_1, v_1, w_2, x_1) & (d_2 = d_{2a}, e_2 = w_2x_1) \\
\wedge\ \mathsf{deleting}(w_2, x_1, w_3, x_3) & (d_2 = d_{2a}, d_3 = w_3x_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, w_3, x_1) & (d_2 = d_{2a}, e_3 = w_3x_1) \\
\wedge\ \mathsf{deleting}(w_3, x_1, w_2, x_2) & (d_2 = d_{2a}, d_4 = w_2x_2) \\
\wedge\ \mathsf{deleting}(w_1, x_1, x_2, w_2) & (d_{2b} = x_2w_2) \\
\wedge\ \mathsf{intersection}(u_1, v_1, x_2, x_1) & (d_2 = d_{2b}, e_2 = x_2x_1) \\
\wedge\ \mathsf{deleting}(x_2, x_1, w_2, x_4) & (d_2 = d_{2b}, d_3 = w_2x_4) \\
\wedge\ \mathsf{intersection}(u_1, v_1, w_2, x_1) & (d_2 = d_{2b}, e_3 = w_2x_1) \\
\rightarrow\ \mathsf{F}(w_2, x_4) & (w_2x_2 \textit{ is not closestF})
\end{array}
$$

If w_2x_4 is in F, this contradicts the condition that w_2x_2 is a closestF edge.



(**a**) The redundancy property is not satisfied for the intersection of the edges w_2x_4 and x_2x_3

(**b**) The edge w_2x_4 is also in F, which is a contradiction to the assumption that w_2x_2 is closestF

Figure 51: In both cases regarded in Step6 a contradiction is obtained

Step6b-c: Next the case that the edge $d_4 = w_4x_4$ deleting x_2x_1 is in F is considered. In this case the vertex w_4 can be assumed to be located inside a triangle formed by w_1, x_1 and x_2. In **Step6b** the vertex w_4 is located inside the triangle $\Delta w_1x_1w_2$, while in **Step6c** the vertex w_4 is located inside the triangle $\Delta x_1w_1w_2$. The edge w_4x_1 is then deleted by an edge $d_5 = w_5x_5$ such that w_5x_1 is intersecting w_2x_2 (**Step6b1** resp. **Step6c1**) or $w_5 = w_2$ (**Step6b2** resp. **Step6c2**). Also in those cases a contradiction is obtained. The proofs are similar to the proofs of Step6a and listed in Appendix 9.4.4.

Existence of an acyclic sequence for deleting edges of the same type Since in all cases of Step6 a contradiction is obtained, the edge x_2x_1 cannot be part of a redundancy cycle that contains the edge w_2x_1 or an edge intersecting with w_2x_2. Therefore, d_2 can be fixed as the edge $d_{2b} = x_2w_2$, leading to $e_2 = x_2x_1$, if the choice of $d_{2a} = w_2x_2$ would lead to a redundancy cycle. This process of choosing an alternative deleting edge is applied, whenever a redundancy cycle of size 2 occurs, i.e. d_i is chosen as x_iw_i instead of w_ix_i. With this choice of d_i, the edge x_ix_1 can therefore not be part of a redundancy cycle that contains the edge w_ix_1 or an edge intersecting with w_ix_i. By choosing the deleting edges in this way, it is guaranteed that no eligible edge is contained twice in the sequence and no deleting edge is chosen twice, before a switching edge is reached. Therefore, it can be assumed that the sequence of eligible edges does not contain a redundancy cycle, where all deleting edges are of the same type.

Sequence of eligible edges between switching edge After proving that no redundancy cycle can exist, where all deleting edges are of the same type, the length of the subsequence between two switching edges is reduced, whenever it is possible, following the proof steps of Part2. Let $e_i = y_{f(i)}z_{g(i)}$ be a switching edge, then according to Step1-5 the following 4 cases can occur for the next switching edge:

- The edge $d_{i+1} = w_{i+1}x_{i+1}$ leads directly to the switching edge $e_{i+1} = w_{i+1}z_{g(i)}$ resp. $e_{i+1} = y_{f(i)}w_{i+1}$.
- The switching edge is $e_j = w_jz_{g(i)}$ resp. $y_{f(i)}w_j$ such that $j \geq i+2$ and w_j is located inside a triangle formed by $y_{f(i)}$, $z_{g(i)}$ ad w_{i+1}.
- The edge $d_{i+2} = w_{i+2}x_{i+2}$ deleting the edge $e_{i+1} = w_{i+1}z_{g(i)}$ resp. $e_{i+1} = y_{f(i)}w_{i+1}$ leads to the switching edge $e_{i+2} = w_{i+2}z_{g(i)}$ resp. $e_{i+1} = y_{f(i)}w_{i+2}$. In this case it can be assumed that $x_{i+2}w_{i+2}$ is not also deleting the edge e_{i+1}, since this leads to one of the first 2 cases according to Step2.
- The switching edge is $e_j = w_jz_{g(i)}$ resp. $y_{f(i)}w_j$ such that $j \geq i+3$ and w_j is located inside a triangle formed by $y_{f(i)}$, w_{i+1} and w_{i+2} resp. $z_{g(i)}$, w_{i+1} and w_{i+2}. Also in this case it can be assumed that $x_{i+2}w_{i+2}$ is not also deleting the edge e_{i+1}.

7.2.3 Proof of Part3: No Redundancy Cycle Among Switching Edges

In this part of the proof a sequence of eligible edges constructed according to the scheme in Fig. 35 by deleting edges of type Y and type Z is considered. Recap that a deleting edge $d_i = w_i x_i$ is of type Y, if the vertex w_i is located on the same side of $u_1 v_1$ as w_1, while d_i is of type Z, if w_i is located on the same side of $u_1 v_1$ as x_1. For such a sequence of eligible edges formed by deleting edges of type Y and deleting edges of type Z, it is proven in the following, that the sequence cannot be cyclic. Since it was already proven that there exist no cyclic sequences containing only deleting edges of type Y resp. only deleting edges of type Z, the focus of this part of the proof is, that the switching edges are not forming a cycle. Switching edges are the edges $y_{2k} z_{2k-1}$, where no deleting edge of type Y is available and the edges $y_{2k} z_{2k+1}$ where no deleting edge of type Z is available. If also the switching edges are not forming a cycle, then there exists an edge $y_{f(t)} z_{g(t)}$ intersecting with $u_1 v_1$ that is not deleted by any other edge and therefore in F, since the graph is finite.

In this part of the proof it has to be distinguished if a deleting edge d_i is of type Y or type Z. The difference between type Y and type Z edges is expressed by using the notation: $d_i(Y)$ if d_i is a deleting edge of type Y resp. $d_i(Z)$ if d_i is a deleting edge of type Z.

In most of the steps of this proof part the following simplification is used: The vertices y_{m_n} are denoted as y_m and the vertices z_{r_s} are denoted as z_r. Following this simplified labeling, the task is now to prove that an eligible edge $y_m z_r$ cannot occur twice in the sequence. Therefore, for an eligible edge $y_{m'} z_{r'}$ with $m' > m$ and $r' > r$ it has to hold that $y_{m'} \neq y_m$ or $z_{r'} \neq z_r$. If the sequence of eligible edges is not further specified, it is assumed that no eligible edges exist between two switching edges.

As a result of Part1 and Part2 of the proof it can be assumed that at least two switching edges $y_2 z_1$ and $y_2 z_3$ exist. The edge $y_2 z_1$ is a switching edge deleting $u_1 v_1$, where all edges deleting $y_2 z_1$ are type Z edges, while the edge $y_2 z_3$ is a switching edge, where all edges deleting $y_2 z_3$ are type Y edges.

If $y_2 z_1$ is considered explicitly as the edge deleting $u_1 v_1$, then for every edge wx deleting $y_2 z_1$, the edge $y_2 w$ is intersecting with $u_1 v_1$. However, in some of the proof parts, $y_2 z_1$ is considered as an arbitrary edge of the sequence that is only intersecting, but not explicitly deleting $u_1 v_1$. Then the condition that for all edges wx deleting the switching edge $y_2 z_1$, the edge $y_2 w$ is intersecting with $u_1 v_1$ or xw also deletes $y_2 z_1$ and $y_2 x$ intersects with $u_1 v_1$ is expressed by adding the axioms $\mathsf{U1} - \mathsf{U2}$:

U1: $\forall w, x : \mathsf{deleting}(y_2, z_1, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_2, w) \vee \mathsf{deleting}(y_2, z_1, x, w)$

U2: $\forall w, x : \mathsf{deleting}(y_2, z_1, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_2, w) \vee \mathsf{intersection}(u_1, v_1, y_2, x)$

The condition that all edges, that delete a switching edge $y_{2k} z_{2k-1}$ are of type Z and all edges, that delete a switching edge $y_{2k} z_{2k+1}$ are of type Y is also added for all other switching edges $y_{2k} z_{2k-1}$ resp. $y_{2k} z_{2k+1}$ considered in the proof. This is expressed by adding the axioms $\mathsf{U3} - \mathsf{U22}$ in the according proof steps:

U3: $\forall w, x : \mathsf{deleting}(y_2, z_3, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_3) \vee \mathsf{deleting}(y_2, z_3, x, w)$
U4: $\forall w, x : \mathsf{deleting}(y_2, z_3, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_3) \vee \mathsf{intersection}(u_1, v_1, x, z_3)$
U5: $\forall w, x : \mathsf{deleting}(y_4, z_3, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_4, w) \vee \mathsf{deleting}(y_4, z_3, x, w)$
U6: $\forall w, x : \mathsf{deleting}(y_4, z_3, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_4, w) \vee \mathsf{intersection}(u_1, v_1, y_4, x)$
U7: $\forall w, x : \mathsf{deleting}(y_4, z_5, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_5) \vee \mathsf{deleting}(y_4, z_5, x, w)$
U8: $\forall w, x : \mathsf{deleting}(y_4, z_5, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_5) \vee \mathsf{intersection}(u_1, v_1, x, z_5)$
U9: $\forall w, x : \mathsf{deleting}(y_6, z_5, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_6, w) \vee \mathsf{deleting}(y_6, z_5, x, w)$
U10: $\forall w, x : \mathsf{deleting}(y_6, z_5, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_6, w) \vee \mathsf{intersection}(u_1, v_1, y_6, x)$
U11: $\forall w, x : \mathsf{deleting}(y_6, z_7, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_7) \vee \mathsf{deleting}(y_6, z_7, x, w)$
U12: $\forall w, x : \mathsf{deleting}(y_6, z_7, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_7) \vee \mathsf{intersection}(u_1, v_1, x, z_7)$
U13: $\forall w, x : \mathsf{deleting}(y_8, z_7, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_8, w) \vee \mathsf{deleting}(y_8, z_7, x, w)$
U14: $\forall w, x : \mathsf{deleting}(y_8, z_7, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_8, w) \vee \mathsf{intersection}(u_1, v_1, y_8, x)$
U15: $\forall w, x : \mathsf{deleting}(y_8, z_9, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_9) \vee \mathsf{deleting}(y_8, z_9, x, w)$
U16: $\forall w, x : \mathsf{deleting}(y_8, z_9, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_9) \vee \mathsf{intersection}(u_1, v_1, x, z_9)$
U17: $\forall w, x : \mathsf{deleting}(y_{10}, z_9, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_{10}, w) \vee \mathsf{deleting}(y_{10}, z_9, x, w)$
U18: $\forall w, x : \mathsf{deleting}(y_{10}, z_9, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_{10}, w) \vee \mathsf{intersection}(u_1, v_1, y_{10}, x)$
U19: $\forall w, x : \mathsf{deleting}(y_{10}, z_{11}, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_{11}) \vee \mathsf{deleting}(y_{10}, z_{11}, x, w)$
U20: $\forall w, x : \mathsf{deleting}(y_{10}, z_{11}, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, w, z_{11}) \vee \mathsf{intersection}(u_1, v_1, x, z_{11})$
U21: $\forall w, x : \mathsf{deleting}(y_{12}, z_{11}, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_{12}, w) \vee \mathsf{deleting}(y_{12}, z_{11}, x, w)$
U22: $\forall w, x : \mathsf{deleting}(y_{12}, z_{11}, w, x) \rightarrow \mathsf{intersection}(u_1, v_1, y_{12}, w) \vee \mathsf{intersection}(u_1, v_1, y_{12}, x)$

Now the proof steps regarding the subsequence of switching edges are described in detail. In Step 1 the case that an edge y_2z_1 is deleting the edge u_1v_1 is regarded. For the Steps2-9 it is enough to consider edges that are intersecting, but not necessarily deleting the edge u_1v_1. The proof steps can therefore be repeated following the flow chart in Fig. 52. In this flow chart, the notation $\overline{\mathsf{inside}}(u, v, w, x)$ is used, which means that x is, depending on the orientation of the triangle, inside the triangle Δuvw or inside the triangle Δvuw. In the proof of Part3 two variables p and q are introduced. The initial values of p and q are set to 1. Whenever a switching vertex y_{2k+2} is located inside a triangle formed by z_{2k-1}, y_{2k} and z_{2k+1} the value of p is set to $2k$. Analogously the value of q is set to $2k+1$, whenever a switching vertex z_{2k+3} is located inside a triangle formed by y_{2k}, z_{2k+1} and y_{2k+2}. The value of p indicates that for $p > 1$ all switching vertices y_{2k} with $2k > p$ are located inside a triangle formed by z_{p-1}, y_p and z_{p+1}, i.e. $\overline{\mathsf{inside}}(z_{p-1}, z_{p+1}, y_p, y_{2k})$ is true. The value of q indicates that for $q > 1$ all switching vertices z_{2k+1} with $2k+1 > q$ are located inside a triangle formed by y_{q-1}, z_q and y_{q+1}, i.e. $\overline{\mathsf{inside}}(y_{q-1}, y_{q+1}, z_q, z_{2k+1})$ is true. The test of all the following proof tasks of Part3 can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/part3

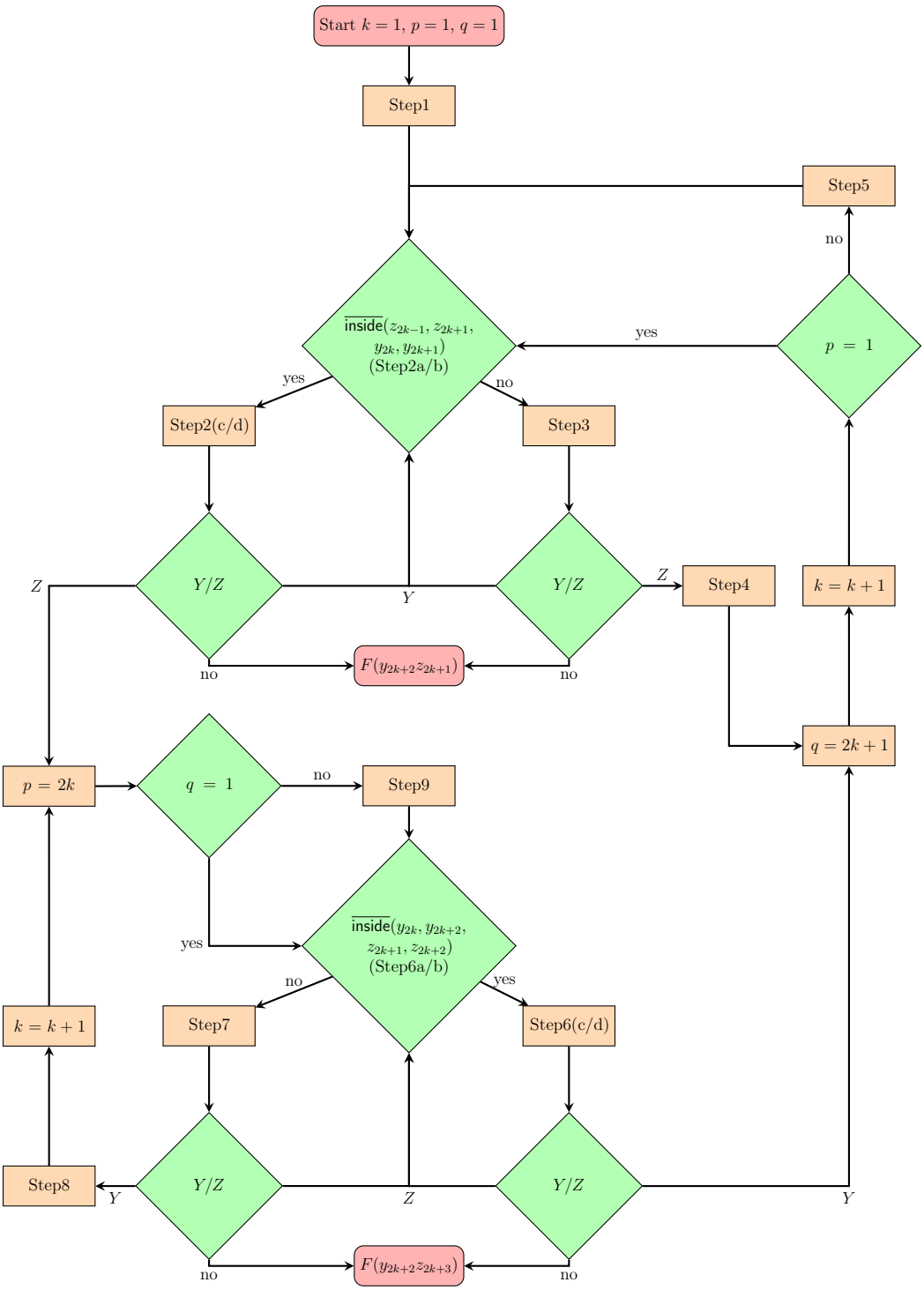


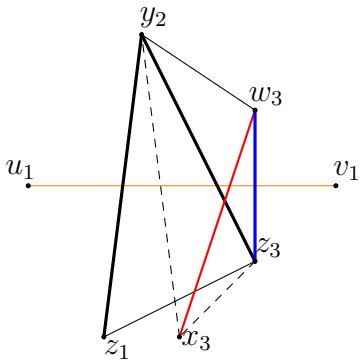
Figure 52: Overview of the proof scheme of Part3

Step1: (see Fig. 53) The switching edge y_2z_1 deletes u_1v_1. Furthermore, the switching edge y_2z_3 is intersecting the edge u_1v_1. The edge z_1z_3 also exists (proven in Part1 Step2), such that the vertices z_1, y_2 and z_3 are forming a triangle.

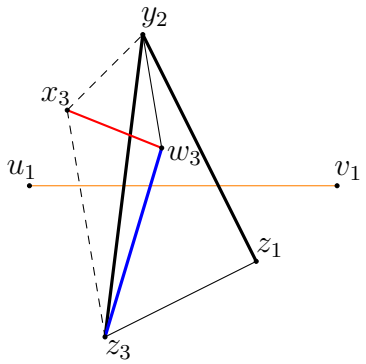
In **Step1a** the edge w_3x_3 deletes y_2z_3. It is proven in **Step1a1** that the vertex w_3 is located inside a triangle formed by z_1, y_2 and z_3, the edge w_3x_3 is in F or w_3x_3 is intersecting with u_1v_1. In **Step1a2** it is furthermore proven that the edges y_2z_1 and w_3z_3 are not intersecting.

In **Step1b** the case that w_3x_3 is in F is considered. It is proven in **Step1b1** that the vertex w_3 is located inside a triangle formed by z_1, y_2 and z_3 or x_3w_3 also deletes y_2z_3. The case that also x_3w_3 deletes y_2z_3 is considered in **Step1b2**, where it is proven that one of the vertices w_3 or x_3 is located inside a triangle formed by z_1, y_2 and z_3. The labeling of the edge w_3x_3 can therefore be chosen such that w_3 is located inside the triangle formed by z_1, y_2 and z_3.

In **Step1c** it is furthermore proven that the vertex z_1 cannot be located inside the triangle $\Delta y_3y_2z_3$. Analogously it is proven in **Step1d** that the vertex z_1 cannot be located inside the triangle $\Delta y_2y_3z_1$. For the proofs of **Step1c** and **Step1c** all cases of the sequence of eligible edges between the switching edges y_2z_1 and y_2z_3 have to be considered separately.



(**a**) The edge w_3x_3 is deleting y_2z_3 and intersecting u_1v_1. Dashed lines represent non-existent edges.



(**b**) The vertex w_3 is located inside the triangle $\Delta z_3z_1y_2$

Figure 53: Illustration of Part3 Step1

Step1a1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{Z1}, \mathbf{U3}-\mathbf{4}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1(Z) = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\rightarrow & \mathsf{F}(w_3, x_3) & (\rightarrow Step1b) \\
\vee & \mathsf{inside}(z_1, z_3, y_2, w_3) \vee \mathsf{inside}(z_1, z_3, y_2, w_3) & (\rightarrow Step2) \\
\vee & \mathsf{intersection}(u_1, v_1, w_3, x_3) & (\rightarrow Step3)
\end{array}
$$

Step1a2: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{Z1}, \mathbf{U3}-\mathbf{4}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1(Y) = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\wedge & \mathsf{intersection}(y_2, z_1, w_3, z_3) & \\
\rightarrow & \bot &
\end{array}
$$

Step1b1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{Z1}, \mathbf{U3}-\mathbf{4}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1(Y) = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_3, z_2, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\wedge & \mathsf{F}(w_3, x_3) & \\
\rightarrow & \mathsf{deleting}(y_3, z_2, x_3, w_3) & (d_{3b}(Y) = x_3w_3(\rightarrow Step1b2)) \\
\vee & \mathsf{inside}(z_1, z_3, y_2, w_3) \vee \mathsf{inside}(z_3, z_1, y_2, w_3) & (\rightarrow Step2)
\end{array}
$$

Step1b2: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{Z1}, \mathbf{U3}-\mathbf{4}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1(Y) = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_3, z_2, w_3, x_3) & (d_{3a}(Y) = w_3x_3) \\
\wedge & \mathsf{deleting}(y_3, z_2, x_3, w_3) & (d_{3b}(Y) = x_3w_3) \\
\rightarrow & \mathsf{inside}(z_1, z_3, y_2, w_3) \vee \mathsf{inside}(z_3, z_1, y_2, w_3) & (\rightarrow Step2) \\
\vee & \mathsf{inside}(z_1, z_3, y_2, x_3) \vee \mathsf{inside}(z_3, z_1, y_2, x_3) & (\rightarrow Step2)
\end{array}
$$

Step1c: The proofs considering all cases of the sequence of eligible edges between the switching edges are listed in Appendix 9.4.5. The result of Step1c(1-6) is the following implication:

$$
\begin{array}{ll}
\mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1(Y) = y_2z_1) \\
\wedge\mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge\mathsf{deleting}(y_2, z_3, y_3, x_3) & (d_3(Y) = y_3x_3) \\
\wedge\mathsf{inside}(y_3, y_2, z_3, z_1) & \\
\rightarrow\perp &
\end{array}
$$

Step1d: The proofs for Step1d(1-6) are similar and are listed in Appendix 9.4.5.

As a result of Step1 the following two cases have to be considered in the further Steps:

- There exists a vertex y_3 inside a triangle formed by z_1, y_2 and z_3, where y_3z_3 is not intersecting with y_2z_1. This case is considered in **Step2**.
- The edge w_3x_3, deleting the edge y_2z_3, is not in F and the vertex w_3 is not located inside a triangle formed by z_1, y_2 and z_3. This case is considered in **Step3**

The following proofs of the Steps2-9 consider a part of a sequence of switching edges $y_2z_1, y_2, z_3, ..., y_{2k-2}z_{2k-1}, y_{2k}z_{2k-1}, y_{2k}z_{2k+1}, y_{2k+2}z_{2k+1}$ and are explicitly given for a fixed value of k. It corresponds to the value of k at the first possible occurrence of the considered step in the scheme in Fig. 52. The proof steps can be repeated for any k, by replacing the indices of all vertices, except for the vertices u_1 and v_1 accordingly.

It is proven that the following properties hold for every k (see Fig. 54):

- If for $j \leq k$ the vertex y_{2j+2} is not located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}, then z_{2k+3} is located inside a triangle formed by y_{2j}, z_{2j+1} and y_{2j+2}. (Step 3 and 4)

- If for $j < k$ the vertex y_{2j+2} is located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1} and z_{2j+3} is located inside a triangle formed by y_{2j}, z_{2j+1} and y_{2j+2}, then the vertex y_{2k+2} is located inside the same triangle formed by z_{2j-1}, y_{2j} and z_{2j+1} as the vertex y_{2j+2}. (Step 2 and 5)

- If for $j < k$ the vertex y_{2j+2} is located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1} and the vertex z_{2j+3} is not located inside a triangle formed by y_{2j}, z_{2j+1} and y_{2j+2}, then the vertex y_{2k+2} is located inside a triangle formed by z_{2j+1}, y_{2j+2} and z_{2j+3} and furthermore located inside the same triangle formed by z_{2j-1}, y_{2j} and z_{2j+1} as the vertex y_{2j+2}. (Step 7 and 8)

- If for $j < k$ the vertex z_{2j+3} is located inside a triangle formed by y_{2j}, z_{2j+1} and y_{2j+2} and the vertex y_{2j+4} is located inside a triangle formed by z_{2j+1}, y_{2j+2} and z_{2j+3}, then the vertex z_{2k+3} is located inside the same triangle formed by y_{2j}, z_{2j+1} and y_{2j+2} as the vertex z_{2j+3}. (Step6 and 9)

Each of the proof Steps2-5 corresponds to a similar proof step in the Steps6-9 and vice versa, where the proofs only differ in the labeling of the vertices and the orientation of the considered triangles. The similar proof steps are Step2 and Step6, Step3 and Step7, Step4 and Step8 as well as Step5 and Step9. For the proofs of the Steps2-5 the orientation of the first triangle that occurs is chosen such that the indices of the considered vertices are increasing. This means a triangle formed by the vertices z_{2k-1}, z_{2k+1} and y_{2k} corresponds to the triangle $\Delta z_{2k+1}z_{2k-1}y_{2k}$ and a triangle formed by the vertices y_{2k}, z_{2k+2} and z_{2k+1} corresponds to the triangle $\Delta y_{2k+2}y_{2k}z_{2k+1}$. In the proofs of the Steps6-9 the other orientation of the first triangle the occurs is considered, i.e. the triangle $\Delta z_{2k-1}z_{2k+1}y_{2k}$ and the triangle $\Delta y_{2k}y_{2k+2}z_{2k+1}$. For all proof steps the proofs for the other orientation of the triangles can be obtained by considering the proof of the similar step with an adapted labeling. Therefore, only the Steps2-5 are described in detail, while the Steps6-9 are listed in Appendix 9.4.5.

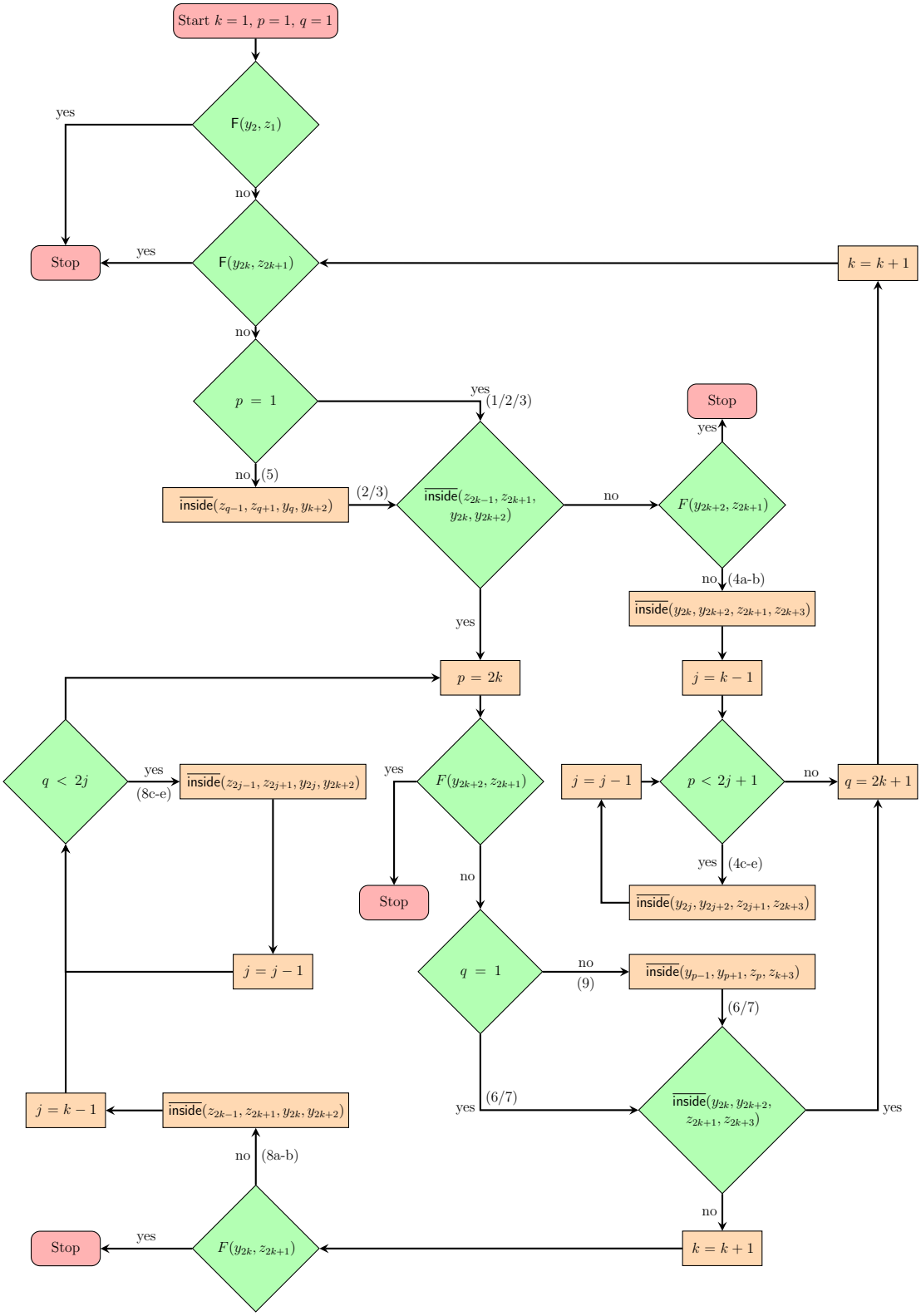


Figure 54: Overview about the proof of Part3, where the numbers in the brackets labeling the arrows correspond to the proof steps of Part3

Step2: The steps **Step2a** and **Step2b** are only relevant for $k \geq 2$, because the case distinction whether a vertex y_{2k+1} is located inside a triangle formed by z_{2k-1}, y_{2k} and z_{2k+1}, is already taken in Step1 for $k = 1$. So the proofs are given explicitly for $k = 2$. For all other values of k the proofs follow by adapting the labeling of the vertices accordingly.

In **Step2a** the vertex z_5 is located inside the triangle $\Delta y_4 y_2 z_3$. In **Step2a1** it is proven that for an edge w_5x_5 deleting y_4z_5, the vertex w_5 is located inside the triangle $\Delta z_5 z_3 y_4$, the edge w_5x_5 is in F, the vertex w_5 is located inside the triangle $\Delta y_4 y_2 z_5$ or the edges w_5z_5 and y_2y_4 are intersecting. It is furthermore proven that the edge w_5z_5 is not intersecting with the edge y_4z_3 (**Step2a2**).

In **Step2b** the case that w_5x_5 is in F is considered. It is proven in **Step2b1** that the vertex w_5 is located inside the triangle $\Delta z_5 z_3 y_4$ or x_5w_5 also deletes y_4z_5. The case that also x_5w_5 deletes y_4z_5 is considered in **Step2b2**, where it is proven that one of the vertices w_5 or x_5 is located inside the triangle $\Delta z_5 z_3 y_4$. The labeling of the edge w_5x_5 can therefore be chosen such that w_5 is located inside the triangle $\Delta z_5 z_3 y_4$.

Step2a1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U3}-\mathbf{8}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4z_5) \\
\wedge & \mathsf{inside}(y_4, y_2, z_3, z_5) & \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5x_5) \\
\rightarrow & \mathsf{F}(w_5, x_5) & (\rightarrow \mathit{Step2b1}) \\
\vee & \mathsf{inside}(z_5, z_3, y_4, w_5) & (\rightarrow \mathit{Step2c}) \\
\vee & \mathsf{inside}(y_4, y_2, z_5, w_5) & (\rightarrow \mathit{Step3}) \\
\vee & \mathsf{intersection}(y_2, y_4, w_5, z_5) & (\rightarrow \mathit{Step3})
\end{array}
$$

The proof for Step2a2 is similar and listed in Appendix 9.4.5.

Step2b1: It is proven that the axioms AxGeom∪AxGraphs∪$\{\mathbf{R1}, \mathbf{C1}\}\cup\{\mathbf{P1}-\mathbf{2}\}\cup\{\mathbf{U3}-\mathbf{8}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4z_5) \\
\wedge & \mathsf{inside}(y_4, y_2, z_3, z_5) & \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5x_5) \\
\wedge & \mathsf{F}(w_5, x_5) & \\
\rightarrow & \mathsf{deleting}(y_4, z_5, x_5, w_5) & (d_{5b}(Y) = x_5w_5(\rightarrow \mathit{Step2b2})) \\
\vee & \mathsf{inside}(z_5, z_3, y_4, w_5) & (\rightarrow \mathit{Step2c})
\end{array}
$$

Step2b2: It is proven that the axioms AxGeom∪AxGraphs∪$\{\mathbf{R1}, \mathbf{C1}\}\cup\{\mathbf{P1}-\mathbf{2}\}\cup\{\mathbf{U3}-\mathbf{8}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4z_5) \\
\wedge & \mathsf{inside}(y_4, y_2, z_3, z_5) & \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_{5a}(Y) = w_5x_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, x_5, w_5) & (d_{5b}(Y) = x_5w_5) \\
\rightarrow & \mathsf{inside}(z_5, z_3, y_4, w_5) & (\rightarrow \mathit{Step2c}) \\
\vee & \mathsf{inside}(z_5, z_3, y_4, x_5) & (\rightarrow \mathit{Step2c})
\end{array}
$$

As a result of Step2a-b the following two cases have to be considered in the further Steps:

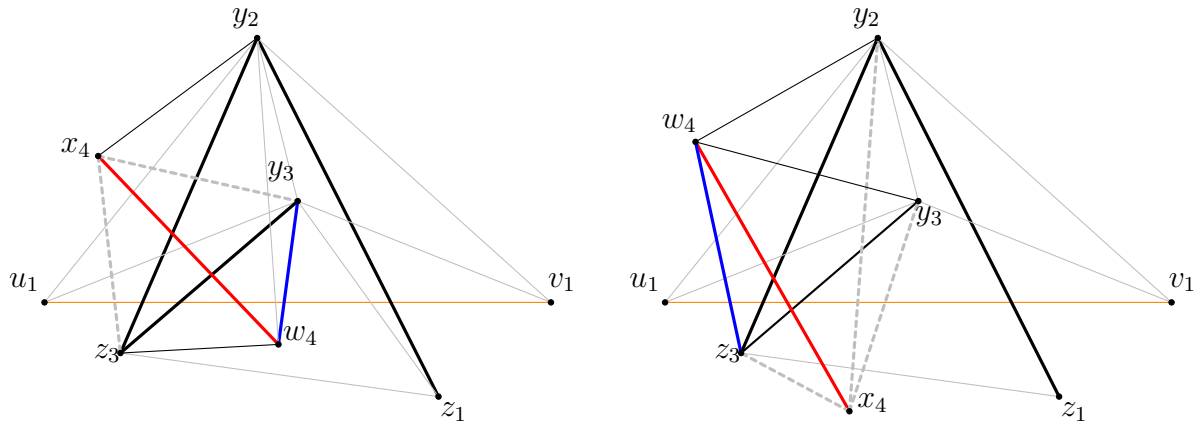
- There exists a vertex y_5 inside a triangle formed by z_3, y_4 and z_5, where y_5z_5 is not intersecting with y_4z_3. This case is considered in **Step2c**.
- The edge w_5x_5, deleting the edge y_4z_5, is not in F and w_5 is not located inside a triangle formed by z_3, y_4 and z_5. This case is considered in **Step3**

For the remaining cases of Step2 a vertex y_{2k+1} is located inside a triangle formed by z_{2k-1}, y_{2k} and z_{2k+1}. This can be a result of Step1 or Step2a-b. Therefore, the proof is explicitly given for $k = 1$, where the vertex y_3 is assumed to be located inside the triangle $\Delta z_3 z_1 y_2$. The edge $y_3 z_3$ is intersecting $u_1 v_1$ and the edge $d_4 = w_4 x_4$ is deleting $y_3 z_3$. If there is a type Y edge available that deletes $y_3 z_3$, this edge is preferred and this proof step can be repeated by considering w_4 instead of y_3 if w_4 is located inside the triangle $\Delta z_3 z_1 y_2$. Type Z edges are therefore only chosen if there is no type Y edge available that deletes $y_3 z_3$. In **Step2c** only the edge $w_4 x_4$ deletes $y_3 z_3$, in **Step2d** also $x_4 w_4$ or $x_3 w_3$ deletes $y_3 z_3$ and in **Step2e** the edge $y_2 x_4$ deletes $y_3 z_3$.

In **Step2c1** (see Fig. 55) it is proven that $w_4 x_4$ is deleting $y_2 z_3$, such that the edge $w_4 x_4$ can be chosen as d_3' instead of $w_3 x_3$, there exists another edge deleting $y_3 z_3$, the vertex w_4 is located inside the triangle $\Delta z_3 z_1 y_2$ or $w_4 x_4$ is an edge of type Z. If $w_4 x_4$ is a type Y edge, it is proven in **Step2c2** that $w_4 z_3$ is not intersecting with $y_2 z_1$.

In **Step2d1** it is proven that one of the vertices w_4 or x_4 is located inside the triangle $\Delta z_3 z_1 y_2$. The labeling can therefore be chosen such that w_4 is located inside the triangle. The case that $w_3 x_3$ and $x_3 w_3$ are deleting $y_2 z_3$ and $w_3 x_3$ also deletes an edge $y_3 z_3$ is considered in **Step2d2**. It is proven that if choosing $d_3 = w_3 x_3$ leads to a redundancy cycle, then there exists another edge deleting $y_2 z_3$ or the edge $y_2 x_4$ is deleting $w_3 z_3$. If $w_4 x_4$ is an edge of type Y it is proven in **Step2d3** that $w_4 z_3$ is not intersecting $y_2 z_1$.

In **Step2e** the sequence of eligible edges between the switching edges $y_2 z_1$ and $y_2 z_3$ has to be considered. In all cases it is proven that is not possible that $y_2 x_4$ deletes $y_3 z_3$, while $x_4 y_2$ is not deleting $y_3 z_3$.



(**a**) w_4 is located inside the triangle $\Delta z_3 z_1 y_2$ and $y_3 w_4$ is intersecting $u_1 v_1$

(**b**) The edge $w_4 x_4$ is deleting $y_2 z_3$ and $w_4 z_3$ is intersecting $u_1 v_1$

Figure 55: Illustration of two possibilities to continue the sequence after Part3 Step2

Step2c1: (see Fig. 55) It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U1}-\mathbf{4}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) & (e_3 = y_3 z_3) \\
\wedge & \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge & \mathsf{deleting}(y_3, z_3, w_4, x_4) & (d_4 = w_4 x_4) \\
\rightarrow & \mathsf{deleting}(y_2, z_3, w_4, x_4) & (d_3'(Y) = w_4 x_4 (\rightarrow Step1/Step2)) \\
\vee & \mathsf{deleting}(y_3, z_3, x_4, w_4) & (d_{4b} = x_4 w_4 (\rightarrow Step2d)) \\
\vee & \mathsf{deleting}(y_3, z_3, y_2, x_4) & (d_4'(Y) = y_2 x_4 (\rightarrow Step2e)) \\
\vee & \mathsf{inside}(z_3, z_1, y_2, w_4) & (d_4(Y) : (\rightarrow Step2c : \ w_4 \ as \ y_3), \\
 & & d_4(Z) : (\rightarrow Step6)) \\
\vee & \mathsf{intersection}(u_1, v_1, y_3, w_4) & (d_4(Z) : (\rightarrow Step6))
\end{array}
$$

The remaining proofs of Step2c-d are similar to the proofs of Step2a-b and listed in Appendix 9.4.5.

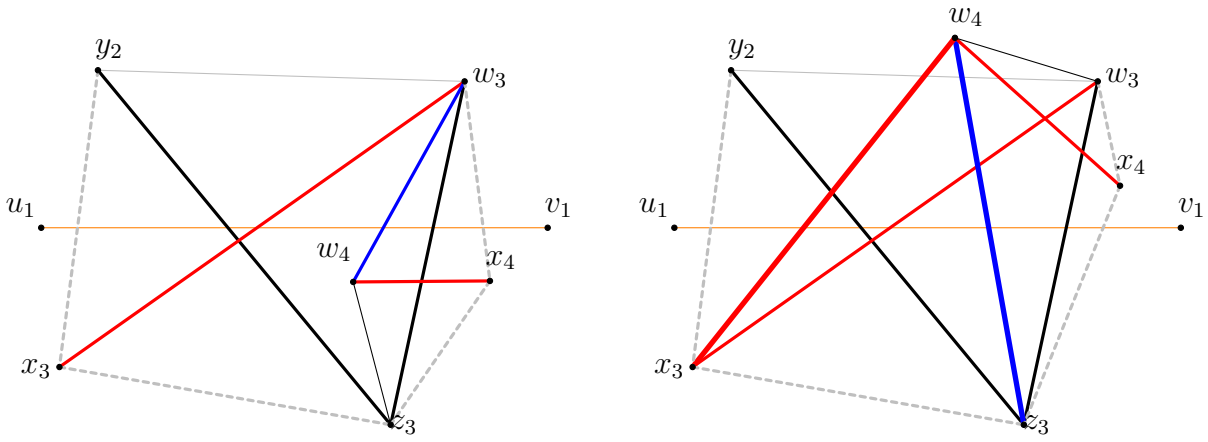
Step2e: The proofs considering all cases of the sequence of eligible edges between the switching edges are listed in Appendix 9.4.5. The result of Step2e(1-6) is the following implication:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) & (e_3 = y_3 z_3) \\
\wedge & \mathsf{deleting}(y_3, z_3, y_2, x_4) & (d_4(Y) = y_2 x_4) \\
\wedge & \neg \mathsf{deleting}(y_3, z_3, x_4, y_2) & (d_{4b}(Y) = x_4 y_2 (Step2d)) \\
\rightarrow & \bot &
\end{array}
$$

As a result of Step2c-e the following two cases have to be considered in the further steps:

- The vertex y_4 is located inside a triangle formed by z_1, y_2 and z_3 This case is considered in **Step6**.
- There exists another edge deleting the edge y_2z_3. This case is considered in **Step1** for $k = 1$ and **Step2** for $k \geq 2$.

Step3: In this step of the proof y_2z_3 is deleted by an edge w_3x_3, which is not in F. Furthermore, an edge y_3z_3 is intersecting u_1v_1. In **Step3a** the case $y_3 = w_3$ and an edge w_4x_4 that deletes w_3z_3 (see Fig. 56) is considered. In **Step3b** the case that a vertex y_3(which can be the vertex w_4 from Step3a) is located inside the triangle $\Delta y_2z_3w_3$ is considered. Furthermore, y_3z_3 and w_3x_3 do not intersect and the edge y_3z_3 is deleted by an edge w_5x_5.



(**a**) w_4 is located inside the triangle $\Delta w_3y_2z_3$ and w_3w_4 is intersecting u_1v_1

(**b**) The edge w_4x_3 is deleting y_2z_3 and w_4z_3 is intersecting u_1v_1

Figure 56: Illustration of two possibilities to continue the sequence after Part3 Step3a

For **Step3a** it is distinguished if only w_4x_4 (**Step3a1** and **Step3a2**) or also x_4w_4 (**Step3a3**, **Step3a4** and **Step3a5**) is deleting y_3z_3. In **Step3a1** it is proven that w_4 is located inside a triangle formed by y_2, z_3 and w_3 or there exists another edge deleting y_2z_3. If there exists another edge deleting y_2z_3 this edge can be considered instead of w_3x_3 in Step1 resp. Step2a. If w_4 is located inside a triangle formed by y_2, z_3 and w_3 it is furthermore proven in **Step3a2** that w_4z_3 and w_3x_3 are not intersecting. If w_4z_3 intersects u_1v_1 the proof is continued in **Step3b** and if y_3w_4 intersects u_1v_1 the proof is continued in Step4. Analogously it is proven in **Step3a3** that one of the vertices w_4 or x_4 is located inside the triangle $\Delta y_2z_3w_3$ or there exists another edge deleting y_2z_3. In the case that one of the vertices w_4 or x_4 is located inside the triangle $\Delta y_2z_3w_3$ the labeling can be chosen such that w_4 is the vertex located inside the triangle. In **Step3a4** it is proven that there exists another edge deleting y_2z_3 if w_3x_3 and y_2w_3 are

intersecting. In **Step3a5** the case that choosing $d_4 = w_4x_4$ leads to a redundancy cycle is considered. In this case another edge deleting y_2z_3 exists or x_4 is also located inside the triangle $\Delta y_2z_3w_3$. If x_4 is also located inside the triangle $\Delta y_2z_3w_3$, the labeling of the vertices is w_4 and x_4 is switched, such that $d_4 = w_4x_4$ does not lead to a redundancy cycle.

Also for **Step3b** it is distinguished if only w_5x_5 (**Step3b1** and **Step3b2**) or also x_5w_5 (**Step3b3** and **Step3b4**) is deleting y_3z_3. In **Step3b1** it is proven that w_5 is located inside the triangle $\Delta y_2z_3w_3$ or there exists another edge deleting y_2z_3. In **Step3b2** it is proven that if the vertex w_5 is located inside the triangle $\Delta y_2z_3w_3$ the edges w_5z_3 and w_3x_3 are not intersecting. Analogously it is proven in **Step3b3** that one of the vertices w_5 or x_5 is located inside the triangle $\Delta y_2z_3w_3$ or there exists another edge deleting y_2z_3. In the case that one of the vertices w_5 or x_5 is located inside the triangle $\Delta y_2z_3w_3$ the labeling can be chosen such that w_5 is the vertex located inside the triangle. In **Step3b4** it is proven that the edges w_5z_3 and w_3x_3 are not intersecting.

So Step3b can be repeated with w_5 as z_3 if w_5x_5 is a deleting edge of type Y or the proof is continued in Step4, if w_5x_5 is a deleting edge of type Z. If there exists another edge deleting y_2z_3 this edge can be considered instead of w_3x_3 in Step1 resp. Step2a.

Step3a1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U3}-\mathbf{4}\}$ entail

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\wedge\neg & \mathsf{F}(w_3, x_3) & (\leftarrow Step1b/2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3z_3) \\
\wedge & \mathsf{deleting}(w_3, z_3, w_4, x_4) & (d_4 = w_4x_4) \\
\rightarrow & \mathsf{deleting}(y_2, z_3, w_4, x_3) & (d_3'(Y) = w_4x_3(\rightarrow Step1/Step2) \\
\vee & \mathsf{deleting}(y_2, z_3, w_4, x_4) & (d_3'(Y) = w_4x_4(\rightarrow Step1/Step2)) \\
\vee & \mathsf{intersection}(w_3, x_3, w_4, z_3) & (\rightarrow Step3a2) \\
\vee & \mathsf{deleting}(w_3, z_3, x_4, w_4) & (d_{4b} = x_4w_4(\rightarrow Step3a3)) \\
\vee & \mathsf{inside}(y_2, w_3, z_3, w_4) \vee \mathsf{inside}(w_3, y_2, z_3, w_4) & (d_4(Y) : (\rightarrow Step3b), \\
 & & d_4(Z) : (\rightarrow Step4a))
\end{array}
$$

The remaining cases of Step3a are similar and listed in Appendix 9.4.5.

Step3b1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U3}-\mathbf{4}\}$ entail

$$
\begin{array}{lr}
\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge \ \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3 x_3) \\
\wedge \neg \mathsf{F}(w_3, x_3) & (\leftarrow Step1b/2b) \\
\wedge \ \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3 z_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) & (e_4 = y_3 z_3) \\
\wedge \ \mathsf{inside}(w_3, y_2, z_3, y_3) & \\
\wedge \neg \mathsf{intersection}(w_3, x_3, y_3, z_3) & (\leftarrow Step3a2) \\
\wedge \ \mathsf{deleting}(y_2, z_3, w_5, x_5) & (d_5 = w_5 x_5) \\
\rightarrow \ \mathsf{deleting}(y_2, z_3, w_5, x_3) & (d_3'(Y) = w_5 x_3 (\rightarrow Step1/Step2)) \\
\vee \ \mathsf{deleting}(y_2, z_3, w_5, x_5) & (d_3'(Y) = w_5 x_5 (\rightarrow Step1/Step2)) \\
\vee \ \mathsf{intersection}(w_3, x_3, w_5, z_3) & (Step3b2) \\
\vee \ \mathsf{deleting}(w_3, z_3, x_5, w_5) & (d_{5b} = x_5 w_5 (\rightarrow Step3b3)) \\
\vee \ \mathsf{inside}(w_3, y_2, z_3, w_5) & (d_5(Y) : (\rightarrow Step3b : w_5 \ as \ y_3, \\
 & \quad d_5(Z) : (\rightarrow Step4b))
\end{array}
$$

The remaining proofs of Step3b are listed in Appendix 9.4.5.

As a result of Step3 the following two cases have to be considered in the further Steps:

- There exists another edge deleting the edge $y_2 z_3$. Then this edge is considered as d_3 instead of $w_3 x_3$ in **Step1** (for $k = 1$) resp. **Step2a** (for $k \geq 2$) .
- The edge $w_3 x_3$ is not in F and deleting the edge $y_2 z_3$. Furthermore, the vertex y_4 is w_3, i.e. $y_4 x_3$ is deleting $y_2 z_3$ or y_4 is located inside a triangle formed by y_2, z_3 and w_3 and the edges $y_4 z_3$ and $w_3 x_3$ are not intersecting. These cases are considered in **Step4**

In the further substeps of Step3 a part of the sequence of switching edges $y_2 z_1, y_2 z_3, ..., y_{2j} z_{2j+1}, y_{2j+2} z_{2j+1}, ..., y_{2k} z_{2k+1}, y_{2k+2} z_{2k+1}, y_{2k+2} z_{2k+3}$ is considered for the case $p < 2j + 1 \leq q$ for $k > j$. It is proven that every eligible edge $y_{2k+2} z_{2k+1}$ is neither intersecting $y_{2j} z_{j+1}$ nor $y_{2j+2} z_{2j+1}$. The proofs are given for fixed values of k and j. For all other values of k and j the proofs follow by adapting the labeling of the vertices accordingly. The proofs in **Step3c** consider the case of $k - j = 1$ and are given explicitly for $k = 2$, while the proofs in **Step3d** consider the case of $k - j = 2$ and are given explicitly for $k = 3$. The proofs of the Steps **Step3e** and **Step3f** consider the cases for $k - j > 2$ are given explicitly for $k - j = 3$ resp. $k - j = 4$. In the proofs of **Step3e** the case of $k - j$

odd is considered and the proofs are given explicitly for $k = 4$ and $j = 1$. In the proofs of **Step3f** the case of $k - j$ even is considered and the proofs are given explicitly for $k = 5$ and $j = 1$.

In **Step3c** it is proven that an edge y_5z_5 cannot intersect with y_2z_3 or y_4z_3. As a result of Step2a the edge $y_{5_1}z_5 = w_5z_5$ is not intersecting with y_4z_3. In **Step3c1** it is proven that w_5z_5 can also not intersect with y_2z_3. So it remains to prove that also a vertex located inside a triangle formed by y_4, w_4 and z_5 is not intersecting with y_2z_3 or y_4z_3. In **Step3c2** it is proven that a vertex y_5 cannot be located inside the triangle $\Delta w_4y_4z_5$ if w_5 is not located inside the triangle $\Delta z_5z_3y_4$. For a vertex y_5 located inside the triangle $\Delta y_4w_5z_5$ it is proven that y_5z_5 is neither intersecting with y_2z_3 (**Step3c3**) nor with y_4z_3 (**Step3c4**). As a result of **Step3c** also the switching edge y_6z_5 can neither intersect with y_2z_3 nor with y_4z_3.

In **Step3d-f** first an edge w_ix_i, deleting the edge $y_{2k}z_{2k+1}$, which is not in F is considered. It is proven that w_iz_{2k+1} is not intersecting with the edge y_2z_3 (**Step3d1**, **Step3e1**, **Step3e2**, **Step3f1**, **Step3f2**). In **Step3d2** it is proven that for a vertex z_{2k+1} located inside the triangle $\Delta y_4y_6z_5$, every edge $y_{2k+1}z_{2k+1}$ that intersects y_4z_3 also intersects y_4z_5. Proving that eligible edges are not intersecting with switching edges of the form $y_{2j}z_{2j+1}$ therefore implies that they are also not intersecting with switching edges of the form $y_{2j+2}z_{2j+1}$.

For an edge w_ix_i, which is not in F and deleting the edge $y_{2k}z_{2k+1}$ it is furthermore proven in **Step3d3** that the vertex y_2 cannot be located inside the triangle $\Delta w_iy_{2k}z_{2k+1}$. In a similar way it is proven in **Step3e3** that the vertex y_4 cannot be located inside the triangle $\Delta y_{2k}w_iz_{2k+1}$.

It remains to consider a vertex y_{2k+1} located inside a triangle formed by w_i, y_{2k}, z_{2k+1}. For such a vertex it is proven that $y_{2k+1}z_{2k+1}$ is not intersecting with y_2z_3 (**Step3d4**, **Step3e4**, **Step3e5**). Therefore, also the switching edge $y_{2k+2}z_{2k+1}$ is not intersecting with the edge y_2z_3. For the switching vertex y_{2k+2}, the transitivity of the inside predicate, implies that y_2 cannot be located inside the triangle $\Delta y_{2k+2}y_{2k}z_{2k+1}$ and y_4 cannot be located inside the triangle $\Delta y_{2k}y_{2k+2}z_{2k+1}$.

Step3c3: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U3}-\mathbf{8}\}$ entail

$$
\begin{array}{lr}
\mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge\ \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge\ \mathsf{intersection}(u_1, v_1, w_5, z_5) \wedge \mathsf{E}(w_5, z_5) & (e_5 = w_5 z_5) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, w_5, z_5) & (\leftarrow Step2a2) \\
\wedge \neg \mathsf{intersection}(y_4, z_3, w_5, z_5) & (\leftarrow Step2a3) \\
\wedge \neg \mathsf{inside}(z_5, z_3, y_4, w_5) & (\leftarrow Step2c) \\
\wedge\ \mathsf{inside}(y_4, w_5, z_5, y_5) & \\
\wedge\ \mathsf{intersection}(y_2, z_3, y_5, z_5) & \\
\rightarrow\ \bot &
\end{array}
$$

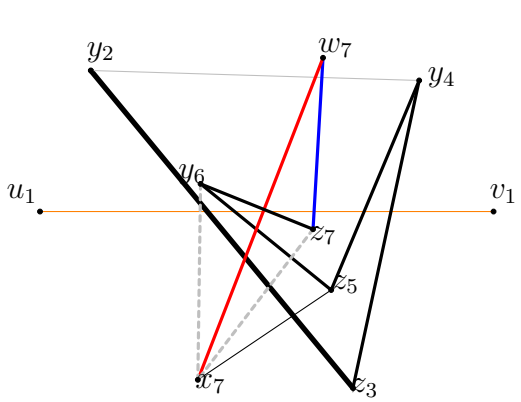
The remaining proofs of Step3c are similar and listed in Appendix 9.4.5.

Step3d1: (see Fig. 57(**a**)) It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U3}-\mathbf{12}\}$ entail

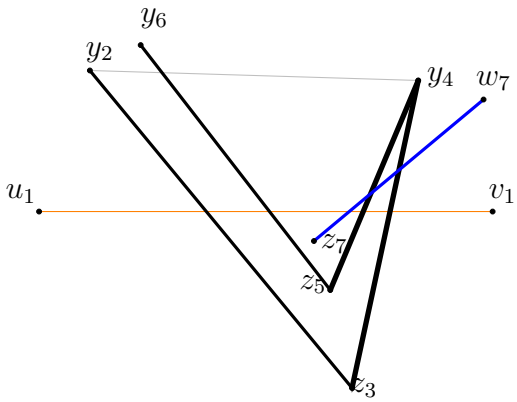
$$
\begin{array}{lr}
\mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) & (e_5 = y_6 z_5) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step2a2\ for\ k-1) \\
\wedge \neg \mathsf{intersection}(y_4, z_3, y_6, z_5) & (\leftarrow Step2a3\ for\ k-1) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) & (e_6 = y_6 z_7) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step4c2\ for\ k-1) \\
\wedge \neg \mathsf{intersection}(y_4, z_3, y_6, z_7) & (\leftarrow Step4c3\ for\ k-1) \\
\wedge\ \mathsf{deleting}(y_6, z_7, w_7, x_7) & (d_7(Y) = w_7 x_7) \\
\wedge \neg \mathsf{F}(w_7, x_7) & (\leftarrow Step1b/2b) \\
\wedge\ \mathsf{intersection}(y_2, z_3, w_7, z_7) & \\
\rightarrow\ \bot &
\end{array}
$$

Step3d2: (see Fig. 57(**b**)) It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U3}-\mathbf{12}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4z_5) \\
\wedge & \mathsf{inside}(y_4, y_2, z_3, z_5) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) & (e_5 = y_6z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_7, z_7) & (e_7 = y_7z_7) \\
\wedge\neg & \mathsf{intersection}(y_4, z_5, y_7, z_7) & (\leftarrow \mathit{Step2a2/Step3\ for\ } j+1) \\
\wedge & \mathsf{intersection}(y_4, z_3, y_7, z_7) & \\
\rightarrow & \bot &
\end{array}
$$



(**a**) The edge w_7y_7 is not intersecting with the edge y_2z_3



(**b**) The edge w_7z_7 is intersecting y_4z_3 and also intersecting y_4z_5.

Figure 57: Illustration of Part3 Step3d

Step3d3: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1-2}\} \cup \{\mathbf{U3-6}, \mathbf{U11-12}\}$ entail:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) && (e_3 = y_4 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_3) && \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) && (e_6 = y_6 z_7) \\
\wedge\ & \mathsf{deleting}(y_6, z_7, w_7, x_7) && (d_7(Y) = w_7 x_7) \\
\wedge & \neg \mathsf{F}(w_7, x_7) && (\leftarrow \mathit{Step2b}) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_7) && (\leftarrow \mathit{Step4c2\ for\ } k-1) \\
\wedge\ & \mathsf{intersection}(y_4, z_3, y_6, z_7) && (\leftarrow \mathit{Step4c3\ for\ } k-1) \\
\wedge\ & \mathsf{inside}(w_7, y_6, z_7, y_2) && \\
\rightarrow\ & \bot &&
\end{aligned}
$$

The remaining proofs of the Steps3d-f are similar and listed in Appendix 9.4.5.

Step4: In this step the sequence of eligible edges after the switching edges y_2z_3 and y_4z_3 is considered. For some of the proofs of Step4a and Step4b it has to be distinguished if the edge y_4x_3 is deleting the edge y_2z_3 (**Step4a1**, **Step4a4**, **Step4b1**, **Step4b4**) or if an edge w_3x_3 is deleting y_2z_3 and the vertex y_4 is located inside a triangle formed by y_2, z_3 and w_3 (**Step4a2**, **Step4a5**, **Step4b2**, **Step4b5**). For the remaining proofs of Step4a and Step4b the edge deleting y_2z_3 is not required and it is therefore enough to consider only the switching edges y_2z_3 and y_4z_3 (**Step4a3**, **Step4a6**, **Step4b3**).

In **Step4a** it is proven that for an edge w_4x_4 resp. w_5x_5 deleting the edge y_4z_3, one of the vertices w_4 or x_4 resp. w_5 or x_5 is located inside the triangle $\Delta y_4y_2z_3$. The case where only w_4x_4 resp. w_5x_5 deletes y_4z_3 is considered in **Step4a1** and **Step4a2**, while the case where also x_4w_4 resp. x_5w_5 deletes y_4z_3 is considered in **Step4a3**, **Step4a4** and **Step4a5**. The labeling of w_4x_4 resp. w_5x_5 can therefore be chosen such that w_4 resp. w_5 is located inside the triangle $\Delta y_4y_2z_3$. In the case that choosing $d_4 = w_4x_4$ resp. $d_5 = w_5x_5$ leads to a redundancy cycle also x_4 resp. x_5 are located inside the triangle $\Delta y_4y_2z_3$. In this case the labeling is adapted such that the choice of $d_4 = w_4x_4$ resp. $d_5 = w_5x_5$ does not lead to a redundancy cycle. Furthermore, it is proven that y_2z_3 and y_4w_4 resp. y_4w_5 are not intersecting (**Step4a6**).

In **Step4b** z_4 is located inside the triangle $\Delta y_4y_2z_3$ and the edge y_4z_4 is intersecting the edge u_1v_1. The edge w_5x_5 resp. w_6x_6 is deleting y_4z_4. Then w_5 resp. w_6 is located inside the triangle $\Delta y_4y_2z_3$ or the edge w_5z_4 resp. w_6z_4 is intersecting with y_2y_4. The

case where only w_5x_5 resp. w_6x_6 deletes y_4z_4 is considered in **Step4b1** and **Step4b2**, while the case where also x_5w_5[8] deletes y_4z_4 is considered in **Step4b3**. If w_5x_5 resp. w_6x_6 is a deleting edge of type Z, then y_4w_5 resp. y_4w_6 are not intersecting with y_2z_3 (**Step4b4** resp. **Step4b5**) and Step4b can be repeated with w_5 resp. w_6 as z_4. If w_5x_5 resp. w_6x_6 is a deleting edge of type Y, then z_5 is located inside the triangle $\Delta y_4y_2z_3$ and the edges y_2z_3 and y_4z_5 are not intersecting. Furthermore, the value of q is set to 3 resp. $2k+1$ for the general case if w_5x_5 resp. w_6x_6 is a deleting edge of type Y.

Step4a1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U3}-\mathbf{6}\}$ entail:

$$
\begin{array}{lr}
\mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge\ \mathsf{deleting}(y_2, z_3, y_4, x_3) & (d_3(Y) = y_4x_3) \\
\wedge\neg\mathsf{F}(y_4, z_3) & (\leftarrow Step1b/2b) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_4, z_3) & (e_3 = y_4z_3) \\
\wedge\ \mathsf{deleting}(y_4, z_3, w_4, x_4) & (d_4(Z) = w_4x_4) \\
\rightarrow\ \mathsf{deleting}(y_4, z_3, x_4, w_4) & (d_{4b} = x_4w_4(\rightarrow Step4a3)) \\
\vee\ \mathsf{inside}(y_2, y_4, z_3, w_4) & (\rightarrow Step4b) \\
\vee\ \mathsf{inside}(y_4, y_2, z_3, w_4) & (\rightarrow Step4b)
\end{array}
$$

Step4a2: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1}-\mathbf{2}\} \cup \{\mathbf{U3}-\mathbf{6}\}$ entail:

$$
\begin{array}{lr}
\mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge\ \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\wedge\neg\mathsf{F}(w_3, z_3) & (\leftarrow Step1b/Step2b) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) & (e_4 = y_4z_3) \\
\wedge\ \mathsf{inside}(w_3, y_2, z_3, y_4) & \\
\wedge\neg\mathsf{intersection}(w_3, z_3, y_4, z_3) & (\leftarrow Step3a2/3b2/3b4) \\
\wedge\ \mathsf{deleting}(y_4, z_3, w_5, x_5) & (d_5(Z) = w_5x_5) \\
\rightarrow\ \mathsf{deleting}(y_4, z_3, x_5, w_5) & (d_{5b} = x_5w_5(\rightarrow Step4a3)) \\
\vee\ \mathsf{inside}(y_4, y_2, z_3, w_5) & (\rightarrow Step4b)
\end{array}
$$

[8]Since the edge deleting y_2z_3 is not required in this proof task, the proof also includes the case where x_6w_6 is deleting y_4z_4

Step4a3: It is proven that the axioms AxGeom∪AxGraphs∪$\{\mathbf{R1}, \mathbf{C1}\}\cup\{\mathbf{P1}-\mathbf{2}\}\cup\{\mathbf{U3}-\mathbf{6}\}$ entail:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) \wedge \neg\mathsf{F}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_4 = y_4z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) & (d_{4a}(Z) = w_4x_4) \\
\wedge & \mathsf{deleting}(y_4, z_3, x_4, w_4) & (d_{4b}(Z) = x_4w_4) \\
\rightarrow & \mathsf{inside}(y_2, y_4, z_3, w_4) \vee \mathsf{inside}(y_4, y_2, z_3, w_4) & (\rightarrow Step4b) \\
\vee & \mathsf{inside}(y_2, y_4, z_3, x_4) \vee \mathsf{inside}(y_4, y_2, z_3, x_4) & (\rightarrow Step4b)
\end{array}
$$

Step4a4: It is proven that the axioms AxGeom∪AxGraphs∪$\{\mathbf{R1}, \mathbf{C1}\}\cup\{\mathbf{P1}-\mathbf{2}\}\cup\{\mathbf{U3}-\mathbf{6}\}$ entail:

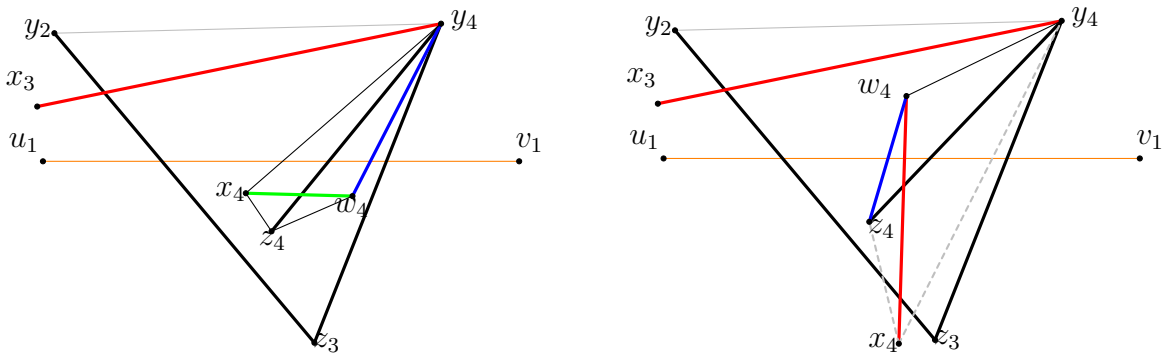
$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, y_4, x_3) & (d_3(Y) = y_4x_3) \\
\wedge & \neg\mathsf{F}(y_4, z_3) & (\leftarrow Step1b/Step2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) & (e_3 = y_4z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) & (d_{4a}(Z) = w_4x_4) \\
\wedge & \mathsf{deleting}(y_4, z_3, x_4, w_4) & (d_{4b}(Z) = x_4w_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_4) \wedge \mathsf{inside}(y_4, y_2, z_3, w_4) & (d_4 = d_{4a}, e_4 = y_4w_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, x_4) & (d_4 = d_{4b}, e_4 = y_4x_4) \\
\wedge & \mathsf{deleting}(y_4, w_4, w_5, x_5) & (d_4 = d_{4a}, d_5(Z) = w_5x_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_5) & (d_4 = d_{4a}, e_5 = y_4w_5) \\
\wedge & \mathsf{deleting}(y_4, w_5, w_4, x_4) & (d_4 = d_{4a}, d_6(Z) = w_4x_4) \\
\rightarrow & \mathsf{inside}(y_4, y_2, z_3, x_4) & (d_4 = d_{4b}(\rightarrow Step4b))
\end{array}
$$

The remaining proofs of Step4a are similar and are listed in Appendix 9.4.5.

Step4b1: (see Fig. 58) It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1-2}\} \cup \{\mathbf{U3-6}\}$ entail:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, y_4, x_3) \qquad (d_3(Y) = y_4x_3) \\
\wedge\neg & \mathsf{F}(y_4, z_3) \qquad (\leftarrow Step1b/Step2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \qquad (e_3 = y_4z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \qquad (e_4 = y_4z_4) \\
\wedge & \mathsf{inside}(y_4, y_2, z_3, z_4) \\
\wedge & \mathsf{deleting}(y_4, z_4, w_5, x_5) \qquad (d_5 = w_5x_5) \\
\rightarrow & \mathsf{deleting}(y_4, z_4, x_5, w_5) \qquad (d_{5b} = x_5w_5(\rightarrow Step4b3)) \\
\vee & \mathsf{inside}(y_4, y_2, z_3, w_5) \qquad (d_5(Z) : (\rightarrow Step4b \ with \ w_5 \ as \ z_4), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad d_5(Y) : (\rightarrow Step2 \ for \ k+1)) \\
\vee & \mathsf{intersection}(y_4, y_2, w_5, z_4) \qquad (d_5(Y) : (\rightarrow Step3 for \ k+1))
\end{array}
$$

The remaining proofs of Step4b are similar to the proofs of Step4a and listed in Appendix 9.4.5.

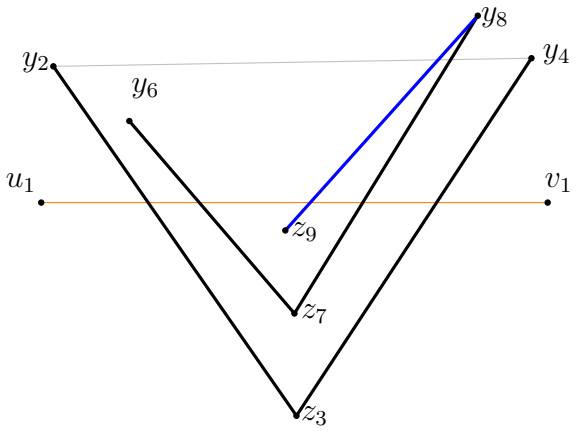


(**a**) The edge y_4w_4 is intersecting u_1v_1 (**b**) The edge w_4z_4 is intersecting u_1v_1

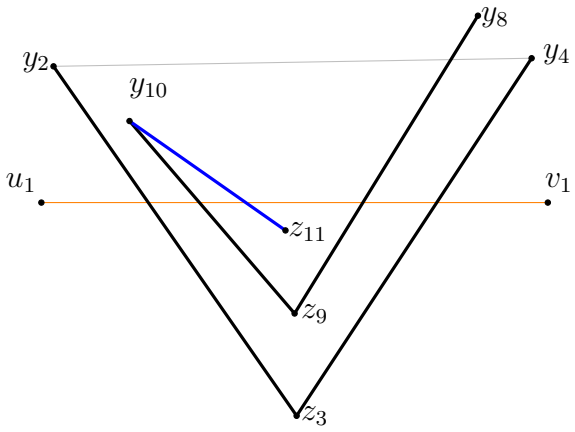
Figure 58: Illustration of two possibilities to continue the sequence after Step4

The further substeps of Step4 a part of the sequence of switching edges $y_2z_1, y_2z_3, ..., y_{2j}z_{2j+1}, y_{2j+2}z_{2j+1}, ..., y_{2k}z_{2k+1}, y_{2k+2}z_{2k+1}, y_{2k+2}z_{2k+3}$ is considered for the case $p < 2j < 2j+1 \leq q$ for $k > j$. The proofs are given for fixed values of k and j. For all other values of k and j the proofs follow by adapting the labeling of the vertices accordingly. The proofs in **Step4c** consider the case of $k - j = 1$ and are given for explicitly $k = 2$ and $j = 1$. The following proofs of the Steps **Step4d** (see Fig. 59(**a**)) and **Step4e** (see Fig. 59(**b**)) consider the cases for $k - j \geq 2$. In the proofs of **Step4d** the case of $k - j$

even is considered and the proofs are given explicitly for $k = 3$ and $j = 1$. In the proofs of **Step4e** the case of $k - j$ odd is considered and the proofs are given explicitly for $k = 4$ and $j = 1$.



(**a**) The vertex z_9 is located inside the triangle $\Delta y_4 y_2 z_3$ and the edge $y_8 z_9$ is not intersecting with one of the edges $y_2 z_3$ or $y_4 z_3$



(**b**) The vertex z_{11} is located inside the triangle $\Delta y_4 y_2 z_3$ and the edge $y_{10} z_{11}$ is not intersecting with one of the edges $y_2 z_3$ or $y_4 z_3$

Figure 59: The vertices z_{2k+1} are located inside the triangle $\Delta y_4 y_2 z_3$ and no switching edge is intersecting a previous switching edge

In the steps **Step4c-e** it is proven that for $k \geq 2$ the vertex z_{2k+1} is located inside the triangle $\Delta y_{2j+2} y_{2j} z_{2j+1}$ (**Step4c1**, **Step4d1**, **Step4e1**) and that the edge $y_{2k} z_{2k+1}$ is not intersecting with one of the edges $y_{2j} z_{2j+1}$ (**Step4c2**, **Step4d2**, **Step4e2**) or $y_{2j+2} z_{2j+1}$ (**Step4c3**, **Step4d3**, **Step4e3**).

Step4c1: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1} - \mathbf{2}\}$ entail:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) && (e_3 = y_4 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) && (e_4 = y_4 z_5) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) && (e_5 = y_6 z_5) \\
\wedge \neg\ & \mathsf{intersection}(y_2, z_3, y_6, z_5) && (\leftarrow \mathit{Step2a2}) \\
\wedge \neg\ & \mathsf{intersection}(y_4, z_3, y_6, z_5) && (\leftarrow \mathit{Step2a3}) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) && (e_6 = y_6 z_7) \\
\rightarrow\ & \mathsf{inside}(y_4, y_2, z_3, z_7)
\end{aligned}
$$

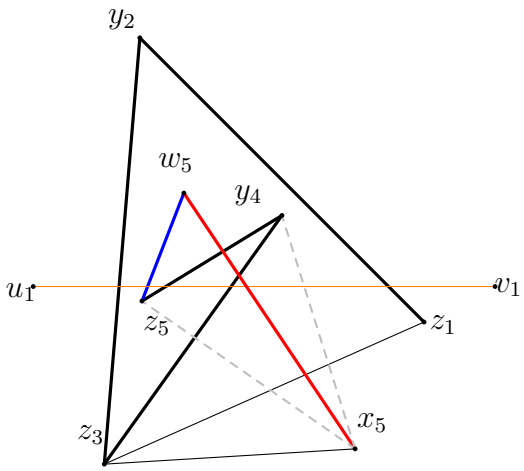
The remaining proofs of the Steps4c-e are similar and are listed in Appendix 9.4.5.

Step5: In this proof step a part of the sequence of switching edges $y_2z_1, y_2z_3, ..., y_{2j}z_{2j-1}$, $y_{2j}z_{2j+1}, y_{2j+2}z_{2j+1}, ..., y_{2k}z_{2k+1}, y_{2k+2}z_{2k+1}, y_{2k+2}z_{2k+3}$ is considered for the case $p = 2j < q$ for $k > j$. The proofs are given for fixed values of k and j. For all other values of k and j the proofs follow by adapting the labeling of the vertices accordingly. The proof in **Step5a** considers the case of $k - j = 1$ and is explicitly given for $k = 2$. In the proof of **Step5b** the case of $k - j = 2$ is considered and the proof is explicitly given for $k = 3$. The following proofs of the Steps **Step5c-d** consider the cases for $k - j > 2$. In the proofs of **Step5c** the case of $k - j$ odd is considered and the proofs are explicitly given for $k = 4$ and $j = 1$. Finally in **Step5d** the case of $k - j$ even is considered and the proof is explicitly given for $k = 5$ and $j = 1$.

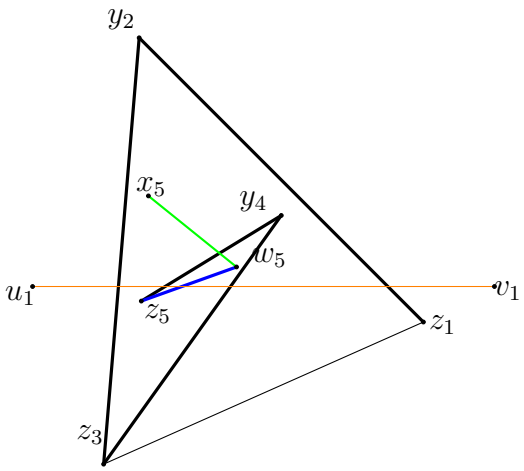
For all cases of this proof step y_4 is located inside the triangle $\Delta z_1z_3y_2$. In **Step5a** (see Fig. 60) the vertex z_5 is located inside the triangle $\Delta z_3y_2y_4$. It is proven that for an edge w_5x_5 deleting y_4z_5 the edge w_5x_5 is in F or w_5 is located inside the triangle $\Delta z_1z_3y_2$. If the edge w_5x_5 is in F, then one of the vertices w_5 or x_5 is located inside the triangle $\Delta z_3z_5y_4$ and therefore also inside the triangle $\Delta z_1z_3y_2$, because of the transitivity of the inside predicate. Using the properties derived in the previous proof steps, it follows that all vertices y_{5_n} and therefore also the switching vertex y_6 are located inside the triangle $\Delta z_1z_3y_2$.

In the further substeps **Step5b-d** of this proof step the vertices y_{2k-2}, z_{2k-1}, y_{2k} and z_{2k+1} are all located inside the triangle $\Delta z_1z_3y_2$. Furthermore, the vertex z_{2k-1} is located inside the triangle $\Delta z_3y_2y_4$ and the vertex z_{2k+1} is located inside a triangle formed by y_{2k-2}, z_{2k-1}, y_{2k}. In addition the switching edges $y_{2k-2}z_{2k-1}$, $z_{2k-1}y_{2k}$ and $y_{2k}z_{2k+1}$ are not intersecting with the edge y_4z_3. In **Step5c** also the edge deleting the switching edge $z_{2k-1}y_{2k}$ has to be considered. Therefore, it has to be distinguished if y_8x_7 deletes y_6z_7 (**Step5c1**) or if an edge w_7x_7 deletes y_6z_7 and y_8 is located inside the triangle $\Delta w_7y_6z_7$(**Step5c2**). For **Step5b** and **Step5d** only the switching edges $y_{2k-2}z_{2k-1}$, $z_{2k-1}y_{2k}$ and $y_{2k}z_{2k+1}$ have to be considered.

In **Step5b**, **Step5c1**, **Step5c2** and **Step5d** it is proven that for an edge w_ix_i deleting $y_{2k}z_{2k+1}$, w_ix_i is in F or w_i is located inside the triangle $\Delta z_1z_3y_2$. Therefore, the vertex y_{2k+2} is located inside the triangle $\Delta z_1z_3y_2$.



(**a**) w_5x_5 is not in F and w_5 is located inside the triangle $\Delta z_3z_1y_2$



(**b**) w_5 is located inside the triangle $\Delta z_5z_3y_4$ and therefore also located inside the triangle $\Delta z_3z_1y_2$

Figure 60: Illustration of two cases considered in Step5 of the proof

Step5a: It is proven that the axioms $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P1-2}\} \cup \{\mathbf{U1-8}\}$ entail:

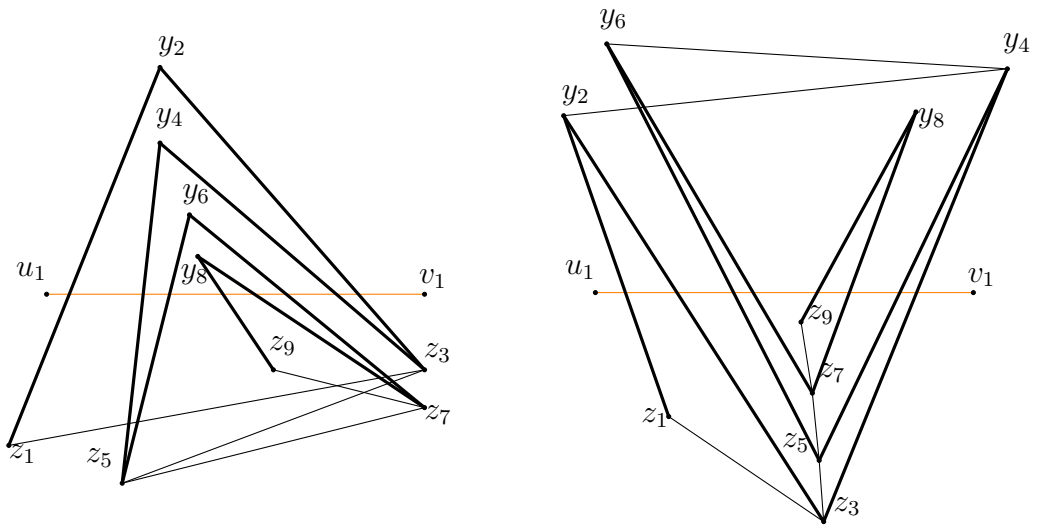
$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_4) & (e_3 = y_4z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5x_5) \\
\rightarrow & \mathsf{F}(w_5, x_5) & (\rightarrow \mathit{Step2b}) \\
\vee & \mathsf{inside}(z_3, z_1, y_2, w_5) &
\end{array}
$$

The proofs of the Steps5b-d are similar and listed in Appendix 9.4.5.

Steps6-9: The proofs of the Steps6-9 are similar to those of the Steps2-5 and are therefore shifted to the Appendix. In Step6 it is proven in Step6a and Step6b that for an edge $w_{2k+2}x_{2k+2}$ in F deleting the edge $z_{2k+1}y_{2k+2}$ one of the vertices w_{2k+2} or x_{2k+2} is located inside a triangle formed by y_{2k}, y_{2k+2} and z_{2k+1}. In Step6c-f the case that z_{2k+2} is located inside a triangle formed by y_{2k}, y_{2k+2} and z_{2k+1} is considered. For an edge $w_{2k+3}x_{2k+3}$ deleting the edge $y_{2k+2}z_{2k+2}$ one of the vertices w_{2k+3} or x_{2k+3} is located inside a triangle formed by y_{2k}, y_{2k+2} and z_{2k+1}, there exists another edge deleting $z_{2k+1}y_{2k+2}$ or the edge $w_{2k+3}z_{2k+2}$ is intersecting the edge $y_{2k}y_{2k+2}$. In Step7 the case that an edge w_ix_i, which is not in F and deleting the edge $z_{2k+1}y_{2k+2}$ is considered. In Step7 it is proven that $z_{2k+3} = w_i$ or z_{2k+3} is located inside a triangle formed by z_{2k+1}, w_i and y_{2k+2}, but $y_{2k+2}z_{2k+3}$ is not intersecting the edge w_ix_i or there exists another edge deleting the edge $z_{2k+1}y_{2k+2}$. In Step8 it is proven that the vertex y_{2k+2} is located inside a triangle formed by z_{2k-1}, z_{2k+1}, y_{2k} if the vertex z_{2k+1} is not located inside a triangle formed by y_{2k-2}, y_{2k} and z_{2k-1}. In Step9 it is proven that if for $j < k$ the vertex z_{2j+3} is located inside a triangle formed by y_{2j}, y_{2j+2} and z_{2j+1} and the vertex y_{2j+4} is located inside a triangle formed by z_{2j+1}, z_{2j+3} and y_{2j+2}, then the vertex z_{2k+3} is located inside the same triangle formed by y_{2j}, y_{2j+2} and z_{2j+1} as z_{2j+3}.

Acyclicity of the sequence of eligible edges: Using the properties for the switching edges derived in the proof steps of Part3 it is now concluded that for $j < k$, where the vertex y_{2j+2} is located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}, the vertex y_{2k+2} is also located inside this triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}. For contradiction it is assumed that this is not the case. Let for an arbitrary, but fixed k, the value of j be chosen maximal, such that y_{2j+2} is located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}, but y_{2k+2} is not located inside this triangle. This means that for all j' with $j < j' < k$, where $y_{2j'+2}$ is located inside a triangle formed by $z_{2j'-1}$, $y_{2j'}$ and $z_{2j'+1}$, also the vertex y_{2k+2} is located inside this triangle (see Fig. 61(**a**)).
First the vertex z_{2j+3} is regarded. If the vertex z_{2j+3} is not located inside a triangle formed by y_{2j}, z_{2j+1} and y_{2j+2}, then by Step8 the vertex y_{2j+4} is located inside a triangle formed by z_{2j+1}, y_{2j+2} and z_{2j+3}. In Step8 it is furthermore proven that for $j < k$ a vertex y_{2k+4} located inside a triangle formed by z_{2j+1}, y_{2j+2} and z_{2j+3} is also located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}. Therefore, a vertex y_{2k+4} that is not located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1} is also not located inside a triangle formed by z_{2j+1}, y_{2j+2} and z_{2j+3}. This contradicts the assumption that j was chosen maximal.

So it can be assumed that the vertex z_{2j+3} is located inside a triangle formed by y_{2j}, z_{2j+1} and y_{2j+2}. In Step5 it is proven that for $p = 2j$ the vertex y_{2k+2} is located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}. Therefore, it can be assumed that $p > 2j$. Then there exists a $j' > j$ such that the vertex $y_{2j'+2}$ is located inside a triangle formed by $z_{2j'-1}$, $y_{2j'}$ and $z_{2j'+1}$, where $y_{2j'}$ is located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}. Furthermore, the edges $y_{2j'}z_{2j'-1}$ and $y_{2j'}z_{2j'+1}$ are not intersecting with one of the edges $y_{2j}z_{2j-1}$ or $y_{2j}z_{2j+1}$.



(**a**) For all k the vertex y_{2k+2} is located inside a triangle formed by z_{2k-1}, y_{2k} and z_{2k+1}

(**b**) For all $k \geq 2$ the vertex y_{2k+1} is located inside a triangle formed by y_{2k-2}, z_{2k-1} and y_{2k}

Figure 61: Illustration of the sequences of edges $y_{2k}z_{2k-1}$ and $y_{2k}z_{2k+1}$ intersecting with u_1v_1

In this case it follows from Step8 that for $j < j' < k$ a vertex y_{2k+2}, located inside a triangle formed by $z_{2j'-1}$, $y_{2j'}$ and $z_{2j'+1}$, is also located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}. Therefore, a vertex y_{2k+2} that is not located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1} is also not located inside a triangle formed by $z_{2j'-1}$, $y_{2j'}$ and $z_{2j'+1}$. This contradicts the assumption that j was chosen maximal.
Now all possible cases are covered and therefore for all j and all $k > j$ it holds that if y_{2j+2} is located inside a triangle formed by z_{2j-1}, y_{2j} and z_{2j+1}, then the vertex y_{2k+2} is also located inside this triangle.

With a similar argumentation it can be concluded that for all j and for all $k > j$ it holds that, if the vertex z_{2j+3} is located inside a triangle formed by y_{2j}, z_{2j+1} and y_{2j+2}, then the vertex z_{2k+3} is also located inside this triangle (see Fig. 61(**b**)).

Therefore, for all k the edges $y_{2k}z_{2k+1}$ and $y_{2k+2}z_{2k+1}$ cannot occur twice in the sequence of eligible edges, which implies that the sequence of switching edges cannot be cyclic.

Putting together the results of Part1, Part2 and Part3 of the proof yields, that there exists an acyclic sequence of eligible edges intersecting with u_1v_1. Since the graph is finite, there exists a t, such that the edge $y_{f(t)}z_{g(t)}$ is intersecting u_1v_1 and no edge is deleting $y_{f(t)}z_{g(t)}$. This edge $y_{f(t)}z_{g(t)}$ is therefore an edge in F that intersects with u_1v_1. □

7.3 Connection Between the Components X and Y

The resulting graph $G' = (V, F)$ after the termination of the algorithm with the components X and $Y = G' \setminus X$ is considered in the following, i.e. every edge in component X resp. Y is an edge in F. Since the components X and Y are assumed to be not connected in $G' = (V, F)$ each edge in F either connects two vertices in X or two vertices in Y, but it is not possible that an edge in F connects a vertex from X with a vertex in Y. While there exists no edge in G' connecting the components X and Y, in the original graph $G = (V, E)$ there exists an edge u_1v_1 with $u_1 \in X$ and $v_1 \in Y$, since G is a connected graph.

7.3.1 Acylicity of the Sequence of Connecting Edges (Lemma 21)

The task considered in the following is to find an edge in E connecting the components X and Y, such that no edge in X intersects with the edge connecting X and Y. For this, a redundancy sequence of edges in E connecting the components X and Y is constructed starting with the edge $e_0 = u_1v_1$. Let for $i \geq 0$ e_i be an edge connecting the components X and Y. If there exists an edge in component X intersecting the edge e_i, then the edge with intersection point closest to v_1 among the edges in X is chosen as $d_{i+1} = w_{i+1}x_{i+1}$. The next edge of the redundancy sequence is $e_{i+1} = w_{i+1}v_1$ Therefore, all edges of this redundancy sequence are incident with v_1. The task is now to show that the redundancy sequence constructed in this way cannot be cyclic. Since only deleting edges in F are relevant and the vertex v_1 is fixed for all edges of the sequence, significantly fewer cases have to be considered as in the proof of Lemma 20. Proving that such a redundancy sequence of edges connecting the components X and Y cannot be cyclic (see Fig. 34(**a**)) yields the following Lemma:

Lemma 21. *There exists an edge uv in E with $u \in X$ and $v \in Y$, such that all edges in F intersecting with uv are located in Y.*

Proof. The edge u_1v_1 is deleted by an edge w_1x_1 in F, such that w_1x_1 is the edge in F in component X with intersection point closest to v_1 among all edges in F in component X that are intersecting with u_1v_1. Then w_1v_1 is an edge connecting the components X and Y.

Step1: For an edge w_2x_2 intersecting v_1w_1, the vertex w_2 is located inside one of the triangles formed by u_1, v_1 and w_1 ($\Delta u_1v_1w_1$ or $\Delta v_1u_1w_1$), x_2w_2 is also deleting v_1w_1 (considered in Step2) or u_1v_1 and w_2x_2 intersect . In case that u_1v_1 and w_2x_2 intersect, this is a contradiction to the assumption that w_1x_1 is the edge in F in component X intersecting with u_1v_1, such that the intersection point p_1 is closest to v_1 among all edges in F in component X intersecting with u_1v_1.

It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P2}\}$ entails:

$$\begin{aligned}
&\mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge \mathsf{F}(w_1, x_1) \\
\wedge&\mathsf{deleting}(w_1, v_1, w_2, x_2) \wedge \mathsf{F}(w_2, x_2) \\
\rightarrow&\mathsf{inside}(u_1, v_1, w_1, w_2) \vee \mathsf{inside}(v_1, u_1, w_1, w_2) \\
\vee&\mathsf{deleting}(v_1, w_1, x_2, w_2) \\
\vee&\mathsf{intersection}(u_1, v_1, w_2, x_2)
\end{aligned}$$

Step2: Next the case that also x_2w_2 is deleting w_1v_1 is considered. Then w_2 or x_2 are located inside one of the triangles formed by u_1, v_1 and w_1 ($\Delta u_1v_1w_1$ or $\Delta v_1u_1w_1$) or u_1v_1 and w_2x_2 intersect . Again an intersection of u_1v_1 and w_2x_2, contradicts the assumpition that w_1x_1 is the edge in F in component X intersecting with u_1v_1, such that the intersection point p_1 is closest to v_1 among all edges in F in component X intersecting with u_1v_1.

It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P2}\}$ entails:

$$\begin{aligned}
&\mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge \mathsf{F}(w_1, x_1) \\
\wedge&\mathsf{deleting}(w_1, v_1, w_2, x_2) \wedge \mathsf{F}(w_2, x_2) \\
\wedge&\mathsf{deleting}(w_1, v_1, x_2, w_2) \\
\rightarrow&\mathsf{inside}(u_1, v_1, w_1, w_2) \vee \mathsf{inside}(v_1, u_1, w_1, w_2) \\
\vee&\mathsf{inside}(u_1, v_1, w_1, x_2) \vee \mathsf{inside}(v_1, u_1, w_1, x_2) \\
\vee&\mathsf{intersection}(u_1, v_1, w_2, x_2)
\end{aligned}$$

Therefore, for the next edge of the sequence connecting the components X and Y, the vertex of component X is located inside a triangle formed by u_1, v_1 and w_1. Since the proofs can be repeated for any construction step of the sequence connecting the components X and Y, this sequence cannot be cyclic if for all i the edge $w_{i+1}x_{i+1}$ in F in component X with intersection point p_{i+1} closest to v_1 is chosen among all edges in F in component X, which are intersecting with w_iv_1. Since the graph is finite, there has to exist an edge w_nv_1, such that w_nv_1 is not intersected by any edge of F or all edges in F intersecting with w_nv_1 are in component Y. □

If w_nv_1 is not intersected by any edge in F, then w_nv_1 is an edge in F connecting the components X and Y. The case that there exists at least one edge in F in component Y intersecting with w_nv_1 is considered in Lemma 22.

7.3.2 Path connecting the components X and Y (Lemma 22)

In this last part of the proof a path containing only edges in F and connecting the components X and Y is derived. This path is constructed by considering the concept of paths along convex

hulls, as in the proof of Lemma 18. Using the results proven in Lemma 20 and Lemma 21, the following Lemma is proven.

Lemma 22. *For an edge u_1v_1 and an intersecting edge $w_1x_1 \in F$, with intersection point p_1 closest to u_1 among all intersecting edges in F, there exists a path P on the convex hull of the triangle $\Delta u_1p_1w_1$ resp. $\Delta u_1w_1p_1$ with all edges in F connecting u_1 and w_1 (see Fig. 62).*

Proof. In order to connect a vertex in X with a vertex in Y, the paths on convex hulls of triangles are regarded. For this, the *finest path* along the convex hull is considered: If vertices c_1, c_2, c_3 are on the convex hull and c_2 is located on the segment c_1c_3 then the edges c_1c_2 and c_2c_3 exist and c_1, c_2, c_3 are all on the finest path of the convex hull (see Fig. 62). In the proof a vertex c_1 located on this path and the vertex c_2, which is the successor of c_1 on this path are considered. Then an edge ab in F cannot intersect the edge c_1c_2.

Step1: First a path on the convex hull of a triangle $\Delta u_1p_1w_1$ formed by an edge u_1v_1 and an intersecting edge $w_1x_1 \in F$ with intersection point p_1 closest to u_1 is regarded (see Fig. 62(**a**)). The first vertex of this path is u_1 and the last vertex of this path is w_1.

If there would exist an edge yz in F intersecting u_1v_1 with intersection point p closer to u_1 then p_1, then one of the edges u_1w_1 or u_1x_1 would be intersected by yz or one of the vertices y or z would be located inside the triangle $\Delta u_1x_1w_1$. Therefore, the following axioms are added to ensure that w_1x_1 is the edge in F with intersection point p_1 closest to u_1 among all edges in F that are intersecting with u_1v_1:

X1: $\forall y, z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(u_1, v_1, y, z) \rightarrow \neg\mathsf{intersection}(u_1, w_1, y, z))$
X2: $\forall y, z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(u_1, v_1, y, z) \rightarrow \neg\mathsf{intersection}(u_1, x_1, y, z))$
X3: $\forall y, z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(u_1, v_1, y, z) \rightarrow \neg\mathsf{inside}(u_1, x_1, w_1, y))$
X4: $\forall y, z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(u_1, v_1, y, z) \rightarrow \neg\mathsf{inside}(u_1, x_1, w_1, z))$

Choosing yz as w_1x_1 does not contradict the axioms $X1-4$ since the edge w_1x_1 is not intersecting with one of the edges u_1w_1 or u_1x_1 and neither w_1 nor x_1 are located inside the triangle $\Delta u_1x_1w_1$.

It is proven that the axioms
$\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P2}\} \cup \{\mathbf{X1-4}, \mathbf{Y1-24}\}$ entail:

$$\begin{aligned} &\mathsf{left}(u_1, v_1, w_1) \wedge \mathsf{left}(v_1, u_1, x_1) \wedge \mathsf{left}(w_1, x_1, v_1) \wedge \mathsf{left}(x_1, w_1, u_1) \\ &\wedge E(u_1, v_1) \wedge F(w_1, x_1) \wedge c_2 = next(c_1) \wedge \mathsf{intersection}(c_1, c_2, a, b) \wedge \mathsf{F}(a, b) \\ &\rightarrow c_1 = nil \vee c_2 = nil \end{aligned}$$

Step2: Next the similar situation for the path on the convex hull of the triangle $\Delta x_1p_1u_1$ from x_1 to u_1 is considered. In this case the vertices are relabeled such that this path

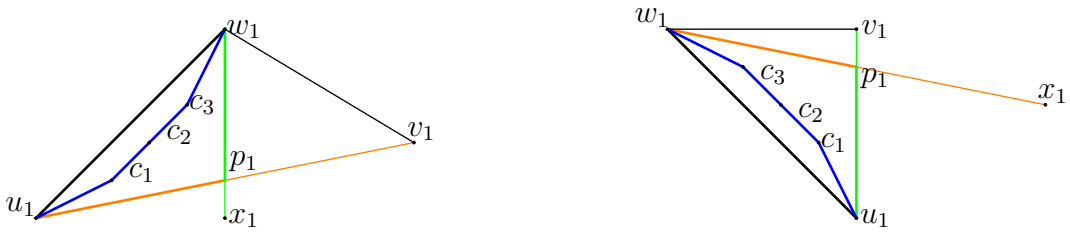
also starts in u_1 and ends in w_1. This can be obtained by replacing the labeling in the following way: u_1 is replaced by w_1, v_1 is replaced by x_1, w_1 is replaced by v_1 and x_1 is replaced by u_1 (see Fig. 62(**b**)). In this case u_1v_1 is the edge in F intersecting w_1x_1 with intersection point q closest to w_1.

The condition that u_1v_1 is the edge in F with intersection point q closest to w_1 among all edges in F intersecting with w_1x_1 is expressed by adding the axioms:

X11: $\forall y, z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(w_1, x_1, y, z) \rightarrow \neg\mathsf{intersection}(u_1, w_1, y, z))$
X12: $\forall y, z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(w_1, x_1, y, z) \rightarrow \neg\mathsf{intersection}(v_1, w_1, y, z))$
X13: $\forall y, z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(w_1, x_1, y, z) \rightarrow \neg\mathsf{inside}(u_1, v_1, w_1, y))$
X14: $\forall y, z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(w_1, x_1, y, z) \rightarrow \neg\mathsf{inside}(u_1, v_1, w_1, z))$

It is proven that the axioms
$\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\} \cup \{\mathbf{P2}\} \cup \{\mathbf{X11-14}, \mathbf{Y1-24}\}$ entail:

$$\begin{aligned}&\mathsf{left}(u_1, v_1, w_1) \wedge \mathsf{left}(v_1, u_1, x_1) \wedge \mathsf{left}(w_1, x_1, v_1) \wedge \mathsf{left}(x_1, w_1, u_1)\\ &\wedge F(u_1, v_1) \wedge E(w_1, x_1) \wedge c_2 = next(c_1) \wedge \mathsf{intersection}(c_1, c_2, a, b) \wedge \mathsf{F}(a, b)\\ &\rightarrow c_1 = nil \vee c_2 = nil\end{aligned}$$



(**a**) Step1: The blue path is connecting the vertices u_1 and w_1

(**b**) Step2: After relabelling the vertices the blue path is connecting the vertices u_1 and w_1

Figure 62: The paths connecting u_1 and w_1

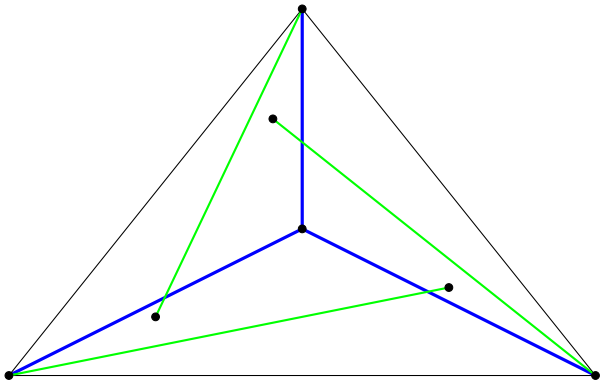
Since the edges of the path on convex hull are not intersected by edges in F, it follows from Lemma 20 that those edges are in F. Therefore, there exists a path with all edges in F that connects u_1 with w_1. □

This path connects the components X and Y and therefore a graph $G' = (V, F)$ resulting from the global CP-algorithm is connected. Since the global and local variant of the CP-algorithm lead to the same output, also the graph resulting from the local CP-algorithm is connected.

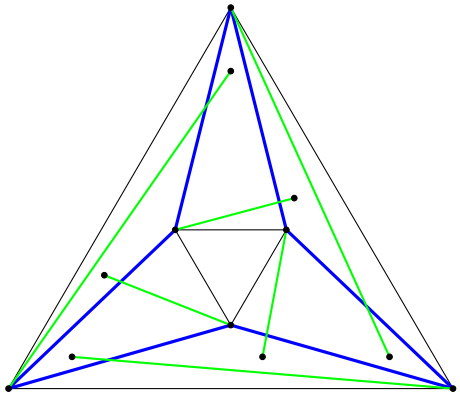
7.4 Constructing of Counterexample Under Weaker Conditions

In this section it is analyzed for certain proof steps if the implications derived in these steps still hold, if some of the assumptions are dropped or weakened.

Analyzing whether the proof still can be carried on if the coexistence property is dropped, resulted in a counterexample, i.e. an example of a geometric graph satisfying axiom **R1** but in which axiom **C1** does not hold, for which no connected plane subgraph can be constructed. The problem occurring for these counterexamples is, that an existential quantification is required for a formula expressing the connectivity of the input graph. So, the obtained formulae might not be in a decidable fragment of first-order logic anymore. However, a counterexample for the case of a redundancy cycle of length 3 can be obtained. The cycle can be constructed by using the predicate deleting. An edge u_1y is deleted by an edge u_2v_2. The redundancy edge u_2y exists, but u_2y is deleted by u_3v_3. This leads to the existence of the redundancy edge u_3y. This edge u_3y is deleted by u_1v_1. Therefore, the edges u_1y, u_2y and u_3y form a redundancy cycle and the graph containing the vertices u_1, u_2, u_3, v_1, v_2, v_3 and y is connected (see Fig. 63(**a**)). It is also possible to construct a redundancy cycle, where each vertex is contained in exactly two edges of the redundancy cycle. Such a cycle of length 6 is illustrated in Fig. 63(**b**). Also for graphs satisfying a weak version of the coexistence property, where only up to two coexistence edges have to exist, the existence of a connected plane subgraph is not guaranteed. All tests for creating the counterexamples, which are considered in the following can be found under https://github.com/lucasboeltz/dissertation-lb/tree/main/part1



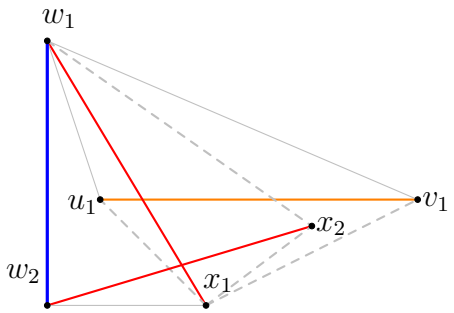
(**a**) The blue edges form a redundancy cycle of length 3



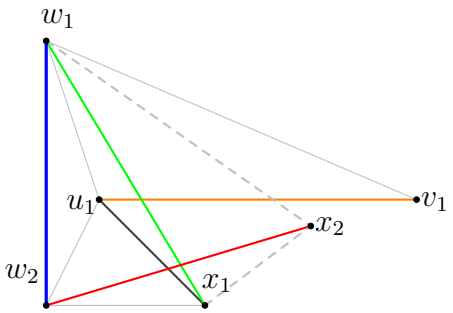
(**b**) The blue edges form a redundancy cycle of length 6

Figure 63: For graphs without the coexistence property the CP-algorithm does not guarantee a connected graph

Next it is analyzed, if for certain proof tasks some of the assumptions can be dropped, such that the considered graph can be simplified, for example by omitting some eligible or deleting edges. In Part1 this is investigated for Step1, where the coexistence property is dropped (see Fig. 64).



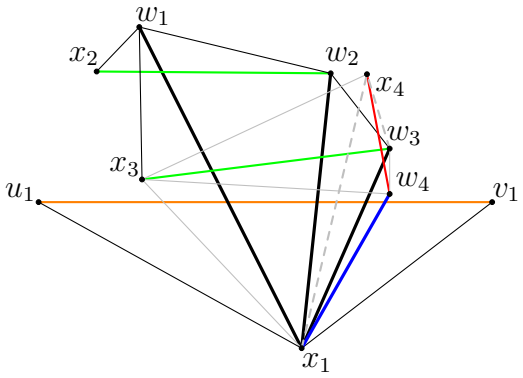
(**a**) Counterexample for Step 1 without coexistence property.



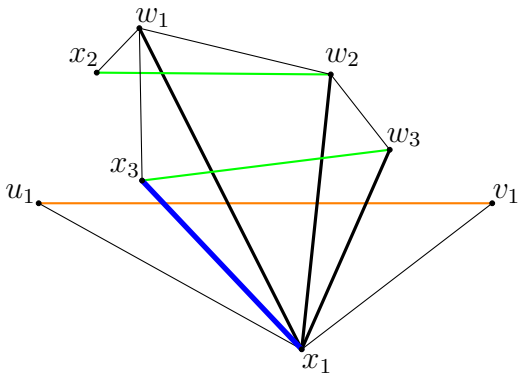
(**b**) With the coexistence property w_1x_1 can only delete u_1v_1 if w_1x_1 is in F

Figure 64: u_1 is located inside the triangle $\Delta w_2x_1w_1$.

Next, the graph considered in Part2 Step5 without the restriction of choosing only the edge in F with intersection point closest to u_1v_1 is analyzed (see Fig. 65).



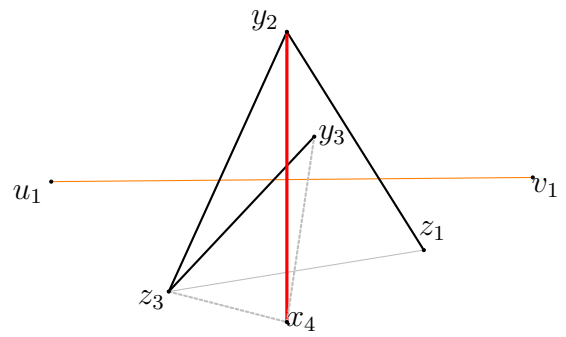
(**a**) No shorter sequence exists if w_3x_3 is chosen as d_3



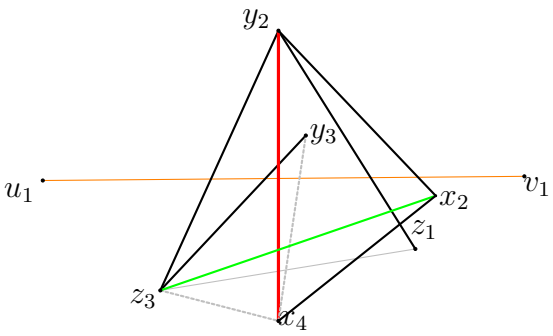
(**b**) The edge x_3w_3 is chosen as d_2

Figure 65: With conflicting deleting edges a sequence containing three eligible edges can be obtained

For the proof steps of Part3 graphs are analyzed, where the deleting edges are omitted and only the eligible edges are included. In Step2e the sequence of eligible edges between the switching edges is omitted (see Fig. 66), in Step3d the edge deleting y_6z_7 is omitted (see Fig. 67) and in Step4d the condition that y_2 is not located inside the triangle $\Delta y_8y_6z_7$ is dropped (see Fig. 68).
It turns out that in all the considered proof tasks a counterexample can be constructed. So the assumptions leading to the counterexample cannot be dropped, in those proof tasks.

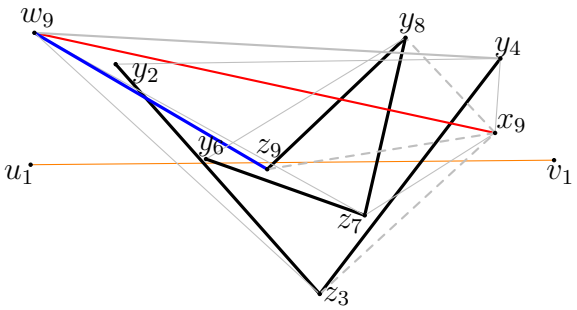


(**a**) The edge y_2x_4 is deleting the edge y_3z_3, but x_4 is not located inside the triangle $\Delta z_3z_1y_2$

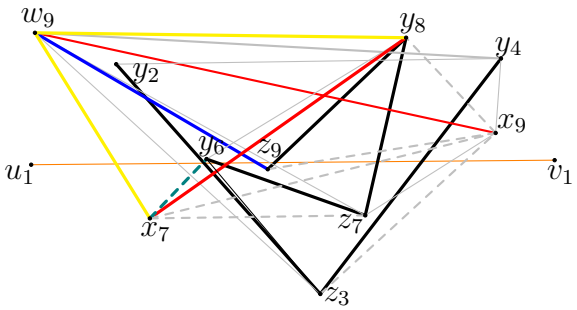


(**b**) Including the edge x_2z_3 leads to a contradiction if the vertex x_4 is not located inside the triangle $\Delta z_3z_1y_2$

Figure 66: The implication of Part3, Step2e can only be proven if the sequence of eligible edges between the switching edges y_2z_1 and y_2z_3 is considered

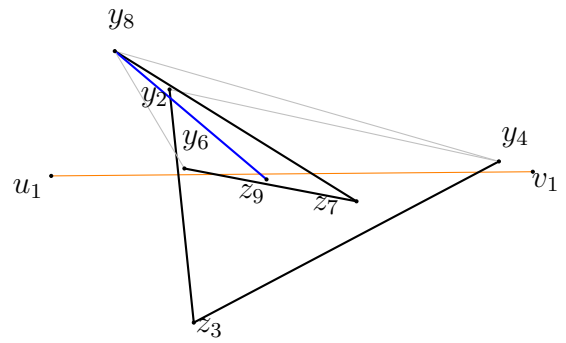


(**a**) The edge w_9z_9 can intersect with y_2z_3 if the edge deleting y_6z_7 is omitted

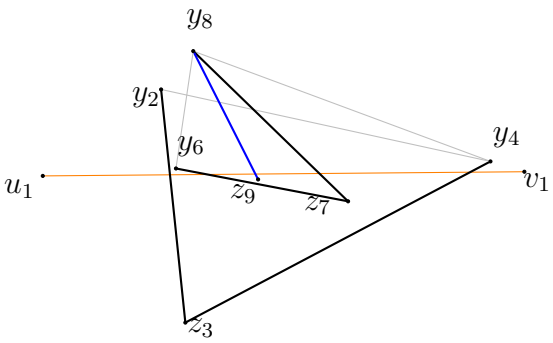


(**b**) If the edge x_7y_8 is included, the coexistence property is not satisfied for the triangle $\Delta y_8w_9x_7$

Figure 67: The implication of Part3, Step3d does only hold if the edge deleting y_6z_7 is included



(**a**) The edge y_8z_9 can intersect the edge y_2z_3 even if y_6z_7 and y_8z_7 are not intersecting y_2z_3



(**b**) The edges y_8z_9 and y_2z_3 cannot intersect, if y_2 is not located inside the triangle $\Delta y_8y_6z_7$

Figure 68: The implication of Part3, Step4 does only hold if y_2 is not located inside the triangle $\Delta y_8y_6z_7$

7.5 Details on the Tests

In this subsection some details of the tests for properties implied by the axiom system, the proof tasks of the CP-algorithm and the tests for constructing counterexamples are presented. The tests were made using the SMT-solver Z3 with Z3-4.8.15 on a 12th Gen Intel(R) Core(TM) i5-12400 CPU @ max 5.6GHz, 18MiB cache. The following tables provide an overview about the running time, the number of vertices and axioms considered in a proof task. A general observation is that the running time is increasing exponentially with the number of vertices, that have to be considered for a proof task. However, also for proof tasks, which consider the same number of vertices, there are still huge differences for the running time. Therefore, the running time of the tests does not only depend on the number of vertices considered in a proof task. One reason for the differences in running times can be found by analyzing the unsat cores. Even if for all proof tasks, the whole axiom system, i.e. $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \mathbf{R1} \cup \mathbf{C1}$ is considered, in some cases a contradiction can already be obtained from the axioms in the query and the axioms in AxGeom. Such proof tasks (e.g. in Part1: Step4, Step8, Step9 and in Part3: Step4c-e) have a significantly shorter running time, than proof tasks, which require also axioms from $\mathsf{AxGraphs} \cup \mathbf{R1} \cup \mathbf{C1}$ to compute an unsat core. Another aspect that significantly affects the runtime of a test is the number of deleting predicates in the query. Since the predicate deleting is defined based on the predicates $\mathsf{intersection}$, E and F, a high number of case distinctions have to be made in the proof tasks containing a huge number of deleting predicates in the query. Consequently, the running time of proof tasks containing a huge number of deleting predicates in the query is quite high (e.g. in Part1: Step6-7, in Part2: Step5, Step6b2, Step6c2 and in Part3: Step3e-f, Step5c.

In Table 13 an overview about the test for properties, derived from the axiom system is given. Afterwards in Tables 14-17 an overview about the tests for verifying the proof steps in the correctness proof of the CP-algorithm is given. Since some of the proof steps are similar to other proof steps, not all proof steps are listed in Tables 14-17. For the proof steps in Part1 and Part3 usually only one of the proof steps, which only differ in the orientation of the considered triangles, is listed. Finally, in Table 18 and overview of the tests for constructing the counterexamples in Section 7.4 is given.

Part	Step	# Clauses Axioms	# Clauses Query	# Constants	Time(s)
Inside	T11	10	3	5	0.08
	T14	10	6	5	0.14
Hilbert	H4(Pasch)	22	12	5	0.51
RCG	Clique	13	9	5	0.34
	Colinear	24	4	5	0.10

Table 13: Details of the test for proving implications of the axiom system

Part	Step	# Clauses Axioms	# Clauses Query	# Constants	Time (s)
closestF	Step1a	41	13	7	3.34
	Step1b	41	10	7	3.4
	Step1c	41	9	6	0.12
	Step2a	43	15	7	2.76
	Step2b	43	12	6	0.11
orderF	Step1	43	6	6	2.19
	Step2	43	9	8	142.82
Part1	Step1a	43	4	6	2.4
	Step1b1	43	3	6	1.02
	Step1b2	44	3	6	0.01
	Step2a1	44	8	8	59.38
	Step2a2	44	8	8	52.37
	Step2a3	44	6	6	2.81
	Step2a4	44	6	6	0.01
	Step2a5	44	11	8	99.27
	Step2a6	44	11	8	75.18
	Step2b1	44	11	8	33.9
	Step2c1	44	12	7	6.95
	Step3a	44	12	7	11.91
	Step3b1	44	11	7	4.01
	Step3b2	44	11	7	3.41
	Step3b3	44	12	7	0.38
	Step3b4	44	13	7	0.34
	Step3b5	44	13	7	0.39
	Step3b6	44	13	7	5.31
	Step4a1	43	7	6	0.89
	Step4a2	43	5	6	0.08
	Step4a3	43	8	6	0.06
	Step4a4	43	8	6	0.07
	Step4a5	43	8	6	0.1
	Step5a	44	10	9	152.5
	Step5b	44	10	9	135.86
	Step6	44	12	10	1115.32
	Step7a	44	14	11	1531.01
	Step7b	44	14	11	1999.62

Table 14: Details of the test for the verification of the proof steps of the CP-algorithm (1 / 4)

Part	Step	# Clauses Axioms	# Clauses Query	# Constants	Time (s)
Part1	Step8a1	43	9	7	1.38
	Step8a2	43	8	7	0.29
	Step8a3	43	8	7	0.26
	Step8a4	43	8	7	0.42
	Step8a5	43	8	7	0.57
	Step9a1	43	8	7	1.41
	Step9a2	43	8	7	0.45
	Step9a3	43	8	7	0.52
Part2	Step1	41	9	6	6.41
	Step2a	41	7	6	2.28
	Step2b	41	8	6	2.61
	Step2c	41	5	6	3.19
	Step3	41	11	6	4.93
	Step4a1	41	10	7	8.9
	Step4a2	41	7	7	1.47
	Step4a3	41	7	7	27.51
	Step4b1	41	10	7	12.29
	Step4b2	41	11	7	3.85
	Step4c1	41	10	7	10.45
	Step4c2	41	7	7	1
	Step4c3	41	7	7	24.86
	Step4d1	41	10	7	10.61
	Step4d2	41	11	7	7.25
	Step5	41	18	8	47.16
	Step6a1	41	5	8	27.56
	Step6a2	41	5	7	7.23
	Step6b1	41	8	10	274.29
	Step6b2	41	7	9	76.13
	Step6c1	41	8	10	317.13
	Step6c2	41	7	9	95.85
Part3	Step1a1	46	9	7	16.79
	Step1a2	46	6	7	12.88
	Step1b1	46	9	7	12.25
	Step1b2	46	10	7	14.31
	Step1c1	46	5	8	27.69
	Step1c2	46	8	9	76.28
	Step1c3	46	8	9	31.88
	Step1c4	46	8	10	377.08
	Step1c5	46	11	11	683.92
	Step1c6	46	11	11	987.78

Table 15: Details of the test for the verification of the proof steps of the CP-algorithm (2 / 4)

Part	Step	# Clauses Axioms	# Clauses Query	# Constants	Time (s)
	Step2a1	49	14	8	16.19
	Step2a2	49	12	8	35.62
	Step2b1	49	13	8	27.09
	Step2b2	49	13	8	35.11
	Step2c1	47	15	8	42.78
	Step2c2	47	14	8	27.2
	Step2d1	47	13	8	43.38
	Step2d2	47	16	10	62.8
	Step2d3	47	15	8	9.91
	Step2e1	47	10	8	19.07
	Step2e2	47	13	9	66.11
	Step2e3	47	13	9	13.08
	Step2e4	47	13	10	340.29
	Step2e5	47	16	11	667.85
	Step2e6	47	16	11	1094.38
	Step3a1	45	13	8	47.74
	Step3a2	45	10	8	35.46
	Step3a3	45	17	10	331.32
	Step3b1	45	16	9	106.15
	Step3b2	45	15	9	62.63
	Step3b3	45	19	9	56.90
	Step3b4	45	17	9	55.46
	Step3c1	49	13	8	12.43
	Step3c2	49	15	8	0.81
	Step3c3	49	16	8	0.47
	Step3c4	49	16	8	0.24
	Step3d1	53	21	10	237.90
	Step3d2	51	17	9	3.95
	Step3d3	49	15	9	51.89
	Step3d4	47	22	10	1.54
	Step3e1	53	25	12	1678.83
	Step3e2	53	28	13	3810.67
	Step3e3	53	23	12	568.06
	Step3e4	53	27	13	3531.28
	Step3e5	53	30	14	9319.24
	Step3f1	53	25	12	2053.43
	Step3f2	53	28	13	3642.27

Table 16: Details of the test for the verification of the proof steps of the CP-algorithm (3 / 4)

Part	Step	# Clauses Axioms	# Clauses Query	# Vertices/Points	Time (s)
	Step4a1	47	10	8	52.43
	Step4a2	47	12	9	130.16
	Step4a3	47	13	7	13.97
	Step4a4	47	14	10	337.63
	Step4a5	47	15	11	771.28
	Step4a6	47	9	7	14.49
	Step4b1	47	13	9	119.90
	Step4b2	47	16	10	345.03
	Step4b3	47	12	8	53.98
	Step4b4	47	12	9	89.20
	Step4b5	47	16	10	241.66
	Step4c1	43	17	8	2.26
	Step4c2	43	17	8	0.53
	Step4c3	43	17	8	0.21
	Step4d1	43	21	9	8.50
	Step4d2	43	21	9	3.62
	Step4d3	43	20	9	0.98
	Step4e1	43	21	9	11.02
	Step4e2	43	21	9	3.93
	Step4e3	43	21	9	4.02
	Step5a	51	15	9	85.91
	Step5b	55	21	11	789.41
	Step5c1	55	29	13	6351.04
	Step5c2	55	32	14	12178.91
	Step5d	55	29	12	2039.12
Part4	Step1	42	8	6	2.21
	Step2	42	9	6	1.47
Part5	Step1	58	5	9	11.55
	Step2	58	5	9	14.12

Table 17: Details of the test for the verification of the proof steps of the CP-algorithm (4 / 4)

Part	Step	# Clauses Axioms	# Constants	# Vertices/Points	Time (s)
cycle	length3	40	7	6	13.43
	double3	40	12	12	3730.76
orderF	orderF	41	9	8	34.92
Part1	Step1	40	4	6	1.36
Part2	Step5	41	18	8	22.16
Part3	Step2e	47	11	7	5.25
Part3	Step3e	53	25	11	389.08
Part3	Step4d	43	20	9	40.09

Table 18: Details of the test for constructing counterexamles

8 Conclusion

In this thesis, different ways to study graphs in the context of WSNs and using methods from automated reasoning for proof tasks and for the construction of models were regarded. First, graphs defined solely by geometric conditions have been investigated, afterwards graphs defined by more general axioms after applying transformations, and finally graphs resulting from a graph transformation algorithm. The contributions of the thesis are presented in the following by referring to the research questions stated in Section 1.

In Section 3 the following research question was considered:

Research question 3: Can methods from automated reasoning be used to test class containedness relations for graph classes related to planarity conditions and for which distance functions is this possible?

The containedness relations between the graph classes Gabriel graph (GG) and relative neighborhood graph (RNG) and the class of plane drawings (P) were analyzed for several distance functions. So far, Gabriel graphs and relative neighborhood graphs seem to be studied only for the Euclidean metric. It was proven that the relation $\mathsf{RNG} \subseteq \mathsf{GG} \subseteq \mathsf{P}$, which was verified for the Euclidean metric ($p = 2$), by using H-PILoT in [13] also holds for all other metrics induced by a p-norm if $p \in (1; \infty)$. However, H-PILoT could not be used to prove this for an arbitrary value of p anymore, since the degrees of considered polynomials become too large for increasing values of p. This is why a manual proof was given instead. Furthermore, the relation between GG, RNG and P was studied, when the distance function is not a metric anymore. For any distance function induced by the p-quasinorm, i.e $p \in (0; 1)$, where the triangle inequality does not hold anymore, it was proven that $\mathsf{GG} \not\subseteq \mathsf{P}$ and also $\mathsf{RNG} \not\subseteq \mathsf{P}$. For concrete quasi-norms it was not possible to prove containedness relations or construct models with H-PILoT, but for a quasi-norm where the triangle inequality was replaced by a weaker condition a counterexample proving that $\mathsf{GG} \not\subseteq \mathsf{P}$ and a counterexample proving that $\mathsf{RNG} \not\subseteq \mathsf{P}$ was constructed by H-PILoT. A task for future work is therefore to develop a method, such that a prover can handle more of the proof tasks. Furthermore, this method, for testing the containedness relations between graph classes, can be considered for further graph classes like the Delaunay triangulation. The Delaunay triangulation was not considered in this thesis, because the condition for the existence of an edge is not clearly determined for graphs with cocircular vertices and requires an existential quantifier.

In the Section 4 the containedness relation of graph classes was considered after applying transformations on a graph class in the following research question:

Research question 2: Under which conditions can class containedness relations between graph classes described by certain predicates under simple transformations be tested by considering only graphs up to a fixed number of vertices?

In Section 4 graph classes defined by inclusion, exclusion and transfer axioms under simple transformations were considered. For such graph classes axiomatizations were derived, in order to obtain conditions for counterexamples, that are contained in one graph class, but not in another graph class. It turned out that for the identity, supergraph, subgraph, reversed graph and empty graph transformation as well as for their negated variants the construction of a counterexample with at most 4 vertices is always possible. For the remaining transformation it was proven that the size of the smallest counterexample can be arbitrary large without further assumptions for the predicates. For the case of the Euclidean distance it was possible to test the containedness relations for all possible graph classes formed by the classes MaxDG, MinDG and CRG and all possible transformation by considering only graphs with up to 4 vertices. The containedness relations between the graph classes where analyzed using Mathematica and a counterexample was constructed if a graph class is not contained in another one. For this approach, the formulae used for checking the containedness conditions by Mathematica were still derived manually, so a question for future work is to analyze for which predicates and which transformations the condition for checking the relation between graph classes can be obtained automatically. A starting point for this task is presented in [61] where by methods of second-order quantifier elimination and hierarchical reasoning the relation $\mathsf{QUDG}^+ = \mathsf{QUDG}^-$ is derived. Furthermore, a task for future work is to consider graph classes, which are described by more general predicates or for more general underlying distance functions.

Another type of graph transformation considered in this thesis were algorithms for the connected planarization of graphs. The possibilities of planarization of graphs was investigated in the following research question:

Research question 1: Under which conditions that are likely to hold in wireless networks the (local) construction of a connected plane subgraph is possible?

For graphs with certain structural properties different possibilities for constructing a connected plane spanning subgraph were considered in Section 5. In Section 5.1 a local approach for the connected planarization of graphs satisfying an acyclic variant of the redundancy property was presented. However, the graphs resulting from this approach are quite sparse. Therefore an orthogonal approach for the connected planarization of graphs in the class RCG was considered. First, the existence of a connected and plane spanning subgraph for graphs in the class RCG was proven in Section 5.2. Afterwards the local CP-algorithm [3] for the connected planarization of graphs in the class RCG was presented in Section 5.3. Since the correctness proof of the CP-algorithm requires many case distinctions, it was considered to verify the proof tasks in a manual derived proof by the theorem prover Z3. In order to use Z3 for the verification of the proof tasks, an axiomatic system modeling a part of the Euclidean geometry was required. The way how parts of Euclidean geometry can be axiomatized was considered in the following research question:

Research question 4: How can the properties for geometric graphs be axiomatized, such that the use of higher-order logic can be avoided, when modeling and verifying proof tasks. Especially, is it possible to handle connectivity without using higher-order logic?

In Section 6 an axiom system, similar to the axiom system of Hilbert and Tarski was defined. With this axiom system, intersection of edges and the property "a point is located inside a triangle formed by 3 other points" can be expressed. Using these geometric properties the conditions for the considered graphs, including the redundancy and the coexistence property can be expressed and finally axioms required for the correctness proof were defined.

The possibility to use the proposed axiom system for the correctness proof of a planarization algorithm was investigated in the following research question:

Research question 5: Can a first-order theorem prover be used to verify the correctness of connected planarization algorithm for graphs satisfying certain structural properties?

In Section 7 the proposed axiom system was used to verify every step of the correctness proof of the CP-algorithm by using the theorem prover Z3. Furthermore, for graphs with a relaxed version of the coexistence property counterexamples were constructed, where the CP-algorithm does not guarantee the output of a connected and plane graph anymore. Also the proof steps were analyzed and it was investigated whether the number of case distinctions can be reduced by dropping some assumptions in the corresponding proof steps. It turned out that the proof cannot be simplified significantly, since otherwise counterexamples can be constructed. A task for future work is to use the axiom system also for proofs concerning other graph classes and axiom systems. However, this is a non-trivial task, since the considered axiom system contains exactly those axiom and the part of Euclidean geometry required for the correctness proof of the CP-algorithm.

In conclusion different possibilities for using automated reasoning for graphs occurring in the study of WSNs were regarded. It was possible to use the methods from automated reasoning in many variants. But for generalizing the proof tasks or develop a proof strategy many manual computations were still required. So there is still a huge way to go for automatizing proofs in geometric graph theory, but using a prover for verifying the proof steps, when the general proof strategy is derived manually, is already a big improvement compared to a completely manual proof.

References

[1] Mohammad Abdulaziz, Kurt Mehlhorn, and Tobias Nipkow. Trustworthy graph algorithms (invited talk). In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 1:1–1:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[2] Lali Barrière, Pierre Fraigniaud, Lata Narayanan, and Jaroslav Opatrny. Robust position-based routing in wireless ad hoc networks with irregular transmission ranges. *Wireless Communications and Mobile Computing*, 3(2):141–153, 2003.

[3] Benjamin Becker. Lokale Algorithmen zur Planarisierung stark redundanter stark koexistenter Graphen. MSC Thesis, University Koblenz-Landau, 2019.

[4] Michael Beeson and Larry Wos. Finding proofs in Tarskian geometry. *J. Autom. Reason.*, 58(1):181–207, 2017.

[5] Rachid Beghdad and Amar Lamraoui. Boundary and holes recognition in wireless sensor networks. *Journal of Innovation in Digital Ecosystems*, 3(1):1–14, 2016. Special issue on Pattern Analysis and Intelligent Systems – With revised selected papers of the PAIS conference.

[6] Douglas M. D.M. Blough, Mauro Leoncini, Giovanni Resta, and Paolo Santi. The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(9):1267–1282, Sep 2006.

[7] Steffen Böhmer, Lucas Böltz, and Hannes Frey. Local Construction of Connected and Plane Spanning Subgraphs under Acyclic Redundancy. In *18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOPT 2020*, 2020.

[8] Steffen Böhmer, Lucas Böltz, and Hannes Frey. WIP: local heuristics for very likely connected and intersection free wireless network topologies under log-normal shadowing. In *23rd IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2022, Belfast, United Kingdom, June 14-17, 2022*, pages 165–168. IEEE, 2022.

[9] Steffen Böhmer, Sven-Niklas Wollny, and Hannes Frey. Locally estimating log-normal shadowing parameters with applications in graph planarization. In Peppino Fazio, Carlos T. Calafate, Danilo Amendola, Eirini-Eleni Tsiropoulou, Maria Diamanti, and Maria Mannone, editors, *15th IFIP Wireless and Mobile Networking Conference, WMNC 2024, Venice, Italy, November 11-12, 2024*, pages 116–123. IEEE, 2024.

[10] Lucas Böltz, Benjamin Becker, and Hannes Frey. Local Construction of Connected Plane Subgraphs in Graphs Satisfying Redundancy and Coexistence. In *XI Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS)*, pages 1–10, 2021.

[11] Lucas Böltz and Hannes Frey. Existence of Connected Intersection-Free Subgraphs in Graphs with Redundancy and Coexistence Property. In *15th International Symposium on Algorithms and Experiments for Wireless Sensor Networks (ALGOSENSORS)*, volume 11931 LNCS, pages 63–78, 2019.

[12] Lucas Böltz and Hannes Frey. Automatically testing containedness between geometric graph classes defined by inclusion, exclusion, and transfer axioms under simple transformations. *Inf.*, 13(12):578, 2022.

[13] Lucas Böltz, Hannes Frey, Dennis Peuter, and Viorica Sofronie-Stokkermans. On testing containedness between geometric graph classes using second-order quantifier elimination and hierarchical reasoning (short paper). In *Proceedings of the Second Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2021) associated with the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021), Online Event, November 4, 2021*, volume 3009 of *CEUR Workshop Proceedings*, pages 37–45. CEUR-WS.org, 2021.

[14] Lucas Böltz, Viorica Sofronie-Stokkermans, and Hannes Frey. On the verification of the correctness of a subgraph construction algorithm. In Rayna Dimitrova, Ori Lahav, and Sebastian Wolff, editors, *25th International Conference on Verification, Model Checking, and Abstract Interpretation*, Lecture Notes in Computer Science. Springer, 2024. To appear.

[15] Prosenjit Bose, Pat Morin, Ivan Stojmenović, and Jorge Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. *Wireless Networks*, 7(6):609–616, nov 2001.

[16] Jean Cardinal, Sébastien Collette, and Stefan Langerman. Local properties of geometric graphs. *Comput. Geom.*, 39(1):55–64, 2008.

[17] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1):165–177, 1990.

[18] Bruno Courcelle. On the expression of monadic second-order graph properties without quantifications over sets of edges (extended abstract). In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pages 190–196. IEEE Computer Society, 1990.

[19] Bruno Courcelle. The monadic second-order logic of graphs VI: on several representations of graphs by relational structures. *Discret. Appl. Math.*, 63(2):199–200, 1995.

[20] Bruno Courcelle. The monadic second-order logic of graphs XVI : Canonical graph decompositions. *Log. Methods Comput. Sci.*, 2(2), 2006.

[21] Bruno Courcelle. Monadic second-order logic for graphs: Algorithmic and language theoretical applications. In Adrian-Horia Dediu, Armand-Mihai Ionescu, and Carlos Martín-Vide, editors, *Language and Automata Theory and Applications, Third International Conference, LATA 2009, Tarragona, Spain, April 2-8, 2009. Proceedings*, volume 5457 of *Lecture Notes in Computer Science*, pages 19–22. Springer, 2009.

[22] David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.)*. Undergraduate texts in mathematics. Springer, 1997.

[23] Leonardo Mendonça de Moura and Nikolaj S. Bjørner. Z3: an efficient SMT solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

[24] Y. Deng and I. Stojmenovic. Partial delaunay triangulations based data-centric storage and routing with guaranteed delivery in wireless ad hoc and sensor networks. In *2009 Mexican International Conference on Computer Science*, pages 24–32, Sept 2009.

[25] David P. Dobkin, Steven J. Friedman, and Kenneth J. Supowit. Delaunay graphs are almost as good as complete graphs. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 20–26. IEEE, 1987.

[26] Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. A new approach for automatic theorem proving in real geometry. *J. Autom. Reason.*, 21(3):357–380, 1998.

[27] Melvin Fitting. *First-Order Logic and Automated Theorem Proving, Second Edition*. Graduate Texts in Computer Science. Springer, 1996.

[28] H. Frey, F. Ingelrest, and D. Simplot-Ryl. Localized minimum spanning tree based multicast routing with energy-efficient guaranteed delivery in ad hoc and sensor networks. In *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–8, June 2008.

[29] Hannes Frey and Lucas Böltz. Automatically testing containedness between geometric graph classes defined by inclusion, exclusion and transfer axioms. In Meng He and Don Sheehy, editors, *Proceedings of the 33rd Canadian Conference on Computational*

Geometry, CCCG 2021, August 10-12, 2021, Dalhousie University, Halifax, Nova Scotia, Canada, pages 127–138, 2021.

[30] K. Ruben Gabriel and Robert R. Sokal. A new statistical approach to geographic variation analysis. *Systematic zoology*, 18(3):259–278, 1969.

[31] Harald Ganzinger. Relating semantic and proof-theoretic concepts for polynominal time decidability of uniform word problems. In *16th Annual IEEE Symposium on Logic in Computer Science, Boston, Massachusetts, USA, June 16-19, 2001, Proceedings*, pages 81–90. IEEE Computer Society, 2001.

[32] Jie Gao, L. J. Guibas, J. Hershberger, Li Zhang, and An Zhu. Geometric spanners for routing in mobile networks. *IEEE Journal on Selected Areas in Communications*, 23(1):174–185, Jan 2005.

[33] Robert Givan and David A. McAllester. New results on local inference relations. In Bernhard Nebel, Charles Rich, and William R. Swartout, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, USA, October 25-29, 1992*, pages 403–412. Morgan Kaufmann, 1992.

[34] Robert Givan and David A. McAllester. Polynomial-time computation via local inference relations. *CoRR*, cs.LO/0007020, 2000.

[35] Andrew V. Goldberg, Serge A. Plotkin, and Gregory E. Shannon. Parallel symmetry-breaking in sparse graphs. *SIAM J. Discret. Math.*, 1(4):434–446, 1988.

[36] David Hilbert. *The Foundations of Geometry*. 2nd ed. Chicago: Open Court., 1980 (1899).

[37] Qingfeng Huang, Chenyang Lu, and G. C. Roman. Reliable mobicast via face-aware routing. In *IEEE INFOCOM 2004*, volume 3, pages 2108–2118 vol.3, March 2004.

[38] Carsten Ihlemann, Swen Jacobs, and Viorica Sofronie-Stokkermans. On local reasoning in verification. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 265–281. Springer, 2008.

[39] Carsten Ihlemann and Viorica Sofronie-Stokkermans. System description: H-PILoT. In Renate A. Schmidt, editor, *Automated Deduction - CADE-22, 22nd International Conference on Automated Deduction, Montreal, Canada, August 2-7, 2009. Proceedings*, volume 5663 of *Lecture Notes in Computer Science*, pages 131–139. Springer, 2009.

[40] Carsten Ihlemann and Viorica Sofronie-Stokkermans. On hierarchical reasoning in combinations of theories. In Jürgen Giesl and Reiner Hähnle, editors, *Automated Reasoning, 5th International Joint Conference, IJCAR 2010, Edinburgh, UK, July 16-19, 2010. Proceedings*, volume 6173 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2010.

[41] Jeny Jacob and Manoj Changat. First-order logic with metric betweenness - the case of non-definability of some graph classes. *AKCE Int. J. Graphs Comb.*, 21(3):232–237, 2024.

[42] Brad Karp and H. T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 243–254, 2000.

[43] Laura Kovács, Hanna Lachnitt, and Stefan Szeider. Formalizing graph trail properties in isabelle/hol. In Christoph Benzmüller and Bruce R. Miller, editors, *Intelligent Computer Mathematics - 13th International Conference, CICM 2020, Bertinoro, Italy, July 26-31, 2020, Proceedings*, volume 12236 of *Lecture Notes in Computer Science*, pages 190–205. Springer, 2020.

[44] F. Kuhn, R. Wattenhofer, and A. Zollinger. An algorithmic approach to geographic routing in ad hoc and sensor networks. *IEEE/ACM Transactions on Networking*, 16(1):51–62, Feb 2008.

[45] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Ad hoc networks beyond unit disk graphs. *Wireless Networks*, 14(5):715–729, Oct 2008.

[46] Harry R. Lewis. Complexity results for classes of quantificational formulas. *J. Comput. Syst. Sci.*, 21(3):317–353, 1980.

[47] X. Li, Y. Deng, V. Narasimhan, A. Nayak, and I. Stojmenovic. Localized address autoconfiguration in wireless ad hoc networks. In *2010 International Conference on Wireless Communications Signal Processing (WCSP)*, pages 1–6, Oct 2010.

[48] Xiang-Yang Li, Gruia Calinescu, and Peng-Jun Wan. Distributed Construction of a Planar Spanner and Routing for Ad Hoc Wireless Networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM '02)*, volume 3, pages 1268–1277, Piscataway, NJ, USA, 2002. IEEE Computer Society.

[49] Nathan Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.

[50] Hamid Mahboubi, Kaveh Moezzi, Amir G. Aghdam, and Kamran Sayrafian-Pour. Distributed sensor coordination algorithms for efficient coverage in a network of heterogeneous mobile sensors. *IEEE Transactions on Automatic Control*, 62(11):5954–5961, 2017.

[51] Tatiana Makarovskikh and Anatoly Panyukov. Special type routing problems in plane graphs. *Mathematics*, 10(5), 2022.

[52] Emi Mathews and Hannes Frey. A Localized Link Removal and Addition Based Planarization Algorithm. In *Proceedings of the 13th International Conference on Distributed Computing and Networking (ICDCN)*, pages 337–350, 2012.

[53] Scott McPeak and George C. Necula. Data structure specifications via local equality axioms. In Kousha Etessami and Sriram K. Rajamani, editors, *Computer Aided Verification, 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005, Proceedings*, volume 3576 of *Lecture Notes in Computer Science*, pages 476–490. Springer, 2005.

[54] N. Mitton, D. Simplot-Ryl, and I. Stojmenovic. Guaranteed delivery for geographical anycasting in wireless multi-sink sensor and sensor-actor networks. In *IEEE INFOCOM 2009*, pages 2691–2695, April 2009.

[55] Kousha Moaveninejad, Wen-Zhan Song, and Xiang-Yang Li. Robust position-based routing for wireless ad hoc networks. *Ad Hoc Networks*, 3(5):546–559, Sep 2005.

[56] Lars Noschinski. A graph library for isabelle. *Mathematics in Computer Science*, 9(1):23–39, 2015.

[57] Lawrence C. Paulson. *Isabelle - A Generic Theorem Prover (with a contribution by T. Nipkow)*, volume 828 of *Lecture Notes in Computer Science*. Springer, 1994.

[58] David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000.

[59] David Peleg and Liam Roditty. Localized spanner construction for ad hoc networks with variable transmission range. *ACM Transactions on Sensor Networks*, 7(3):1–14, Oct 2010.

[60] Dennis Peuter. *Applications for Symbol Elimination in Combination with Hierarchical Reasoning*. PhD thesis, University of Koblenz and Landau, Germany, 2024.

[61] Dennis Peuter and Viorica Sofronie-Stokkermans. Symbol elimination and applications to parametric entailment problems. In Boris Konev and Giles Reger, editors, *Frontiers of Combining Systems - 13th International Symposium, FroCoS 2021, Birmingham, UK,*

September 8-10, 2021, Proceedings, volume 12941 of *Lecture Notes in Computer Science*, pages 43–62. Springer, 2021.

[62] Sumesh J Philip, Joy Ghosh, Hung Q Ngo, and Chunming Qiao. Routing on Overlay Graphs in Mobile Ad Hoc Networks. In *Proceedings of the IEEE Global Communications Conference, Exhibition & Industry Forum (GLOBECOM'06)*, 2006.

[63] Ruzica Piskac, Leonardo Mendonça de Moura, and Nikolaj S. Bjørner. Deciding effectively propositional logic using DPLL and substitution sets. *J. Autom. Reason.*, 44(4):401–424, 2010.

[64] Sylvia Rathnasamy, Brad Krap, Scott Shenker, Deborah Estrin, Ramesh Govindan, Li Yin, and Fang Yu. Data-centric storage in sensornets with ght, a geographic hash table. *Mobile Networks and Applications*, 8(4):427–442, Aug 2003.

[65] Juan A. Sanchez, Pedro M. Ruiz, Jennifer Liu, and Ivan Stojmenovic. Bandwidth-Efficient Geographic Multicast Routing Protocol for Wireless Sensor Networks. *IEEE Sensors Journal*, 7(5):627–636, may 2007.

[66] Nico Schröder. Visualisierung von axiomatisch beschriebenen graphengegenbeispielen im kontext lokaler verfahren in drahtlosnetzmodellen. Bachelor's thesis, Universität Koblenz, Deutschland, 2025.

[67] Mahtab Seddigh, Julio Solano González, and Ivan Stojmenovic. Rng and internal node based broadcasting algorithms for wireless one-to-one networks. *SIGMOBILE Mobile Computing and Communications Review*, 5(2):37–44, April 2001.

[68] Valerie Simon. Automatisches beweisen in der geometrischen graphentheorie. Bachelor's thesis, Universität Koblenz, Deutschland, 2024.

[69] Viorica Sofronie-Stokkermans. Hierarchic reasoning in local theory extensions. In Robert Nieuwenhuis, editor, *Proc. 20th International Conference on Automated Deduction (CADE 20)*, LNCS 3632, pages 219–234. Springer, 2005.

[70] I. Stojmenovic. Geocasting with guaranteed delivery in sensor networks. *IEEE Wireless Communications*, 11(6):29–37, Dec 2004.

[71] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, 2002.

[72] Thomas Sturm and Volker Weispfenning. Computational geometry problems in REDLOG. In Dongming Wang, editor, *Automated Deduction in Geometry, International Workshop*

on Automated Deduction in Geometry, Toulouse, France, September 27-29, 1996, Selected Papers, volume 1360 of *Lecture Notes in Computer Science*, pages 58–86. Springer, 1996.

[73] Jindong Tan. *A Scalable Graph Model and Coordination Algorithms for Mobile Sensor Networks*. Springer US, Boston, MA, 2008.

[74] Alfred Tarski and Steven Givant. Tarski's system of geometry. *Bull. Symb. Log.*, 5(2):175–214, 1999.

[75] Godfried T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.

[76] Hua-Wen Tsai, Chih-Ping Chu, and Tzung-Shi Chen. Mobile object tracking in wireless sensor networks. *Computer Communications*, 30(8):1811–1825, June 2007.

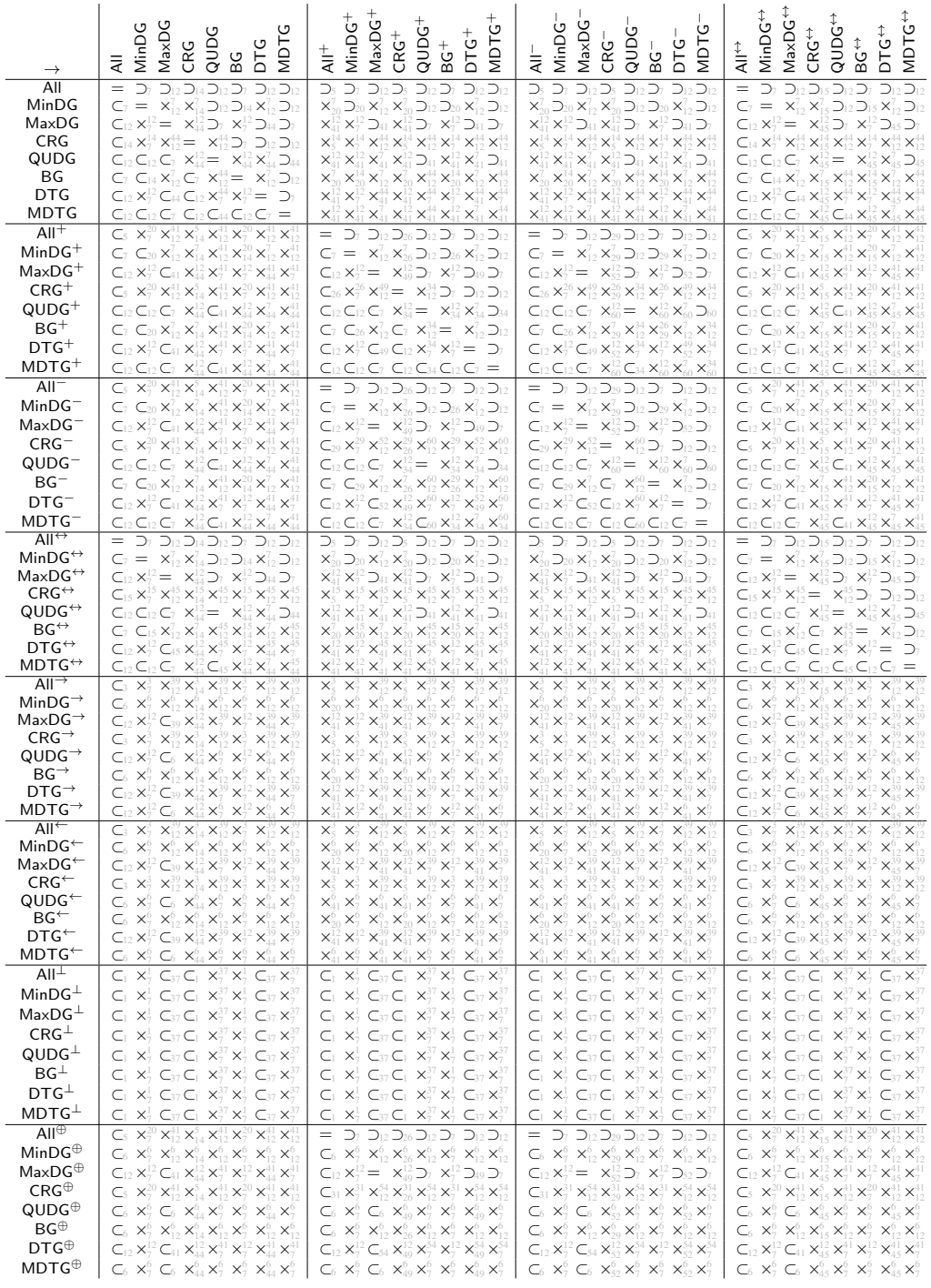
[77] Pascal von Rickenbach, Roger Wattenhofer, and Aaron Zollinger. Algorithmic models of interference in wireless ad hoc and sensor networks. *IEEE/ACM Transactions on Networking*, 17(1):172–185, Feb 2009.

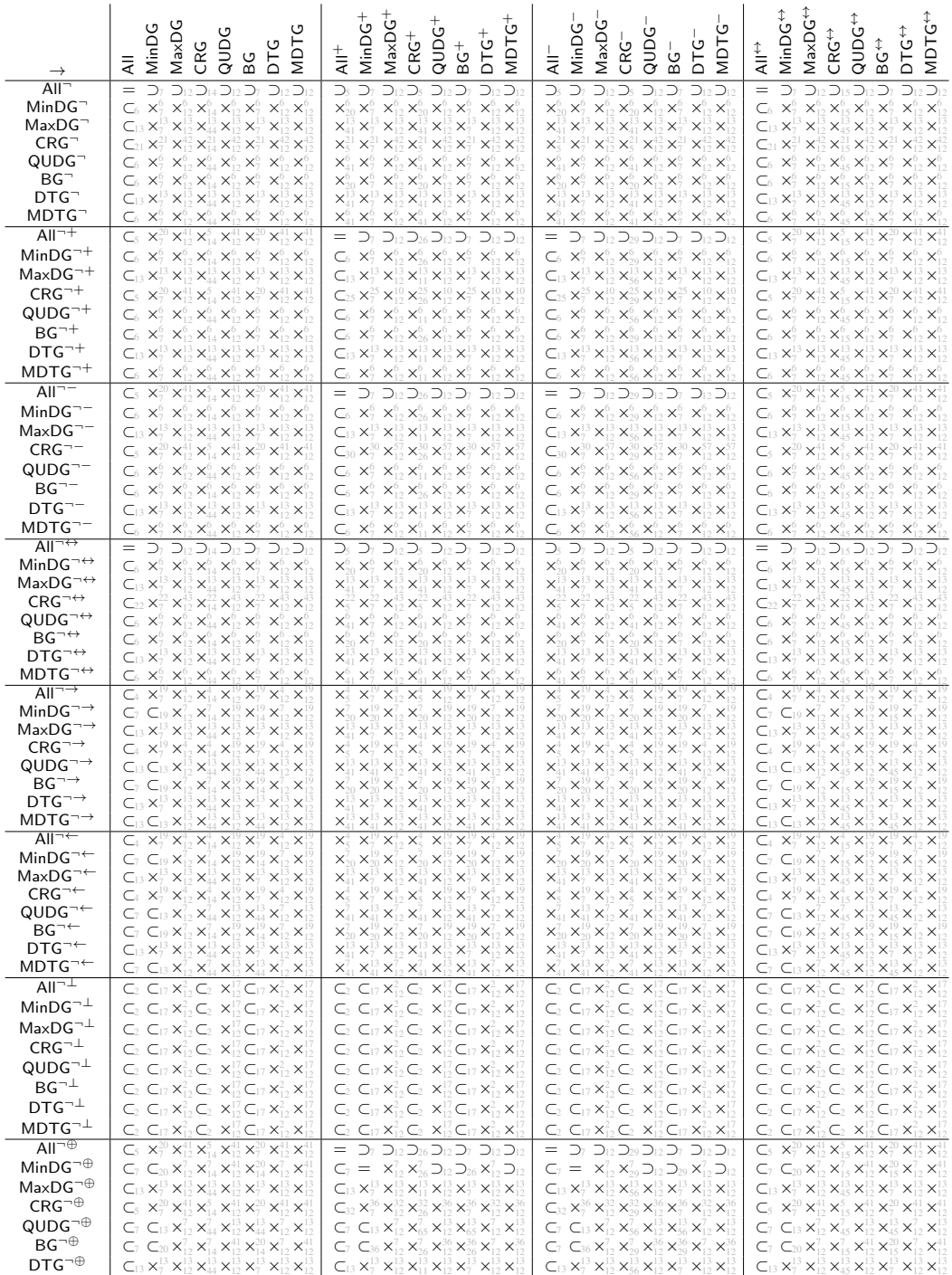
[78] Christoph Weidenbach, Dilyana Dimova, Arnaud Fietzke, Rohit Kumar, Martin Suda, and Patrick Wischnewski. SPASS version 3.5. In Renate A. Schmidt, editor, *Automated Deduction - CADE-22, 22nd International Conference on Automated Deduction, Montreal, Canada, August 2-7, 2009. Proceedings*, volume 5663 of *Lecture Notes in Computer Science*, pages 140–145. Springer, 2009.

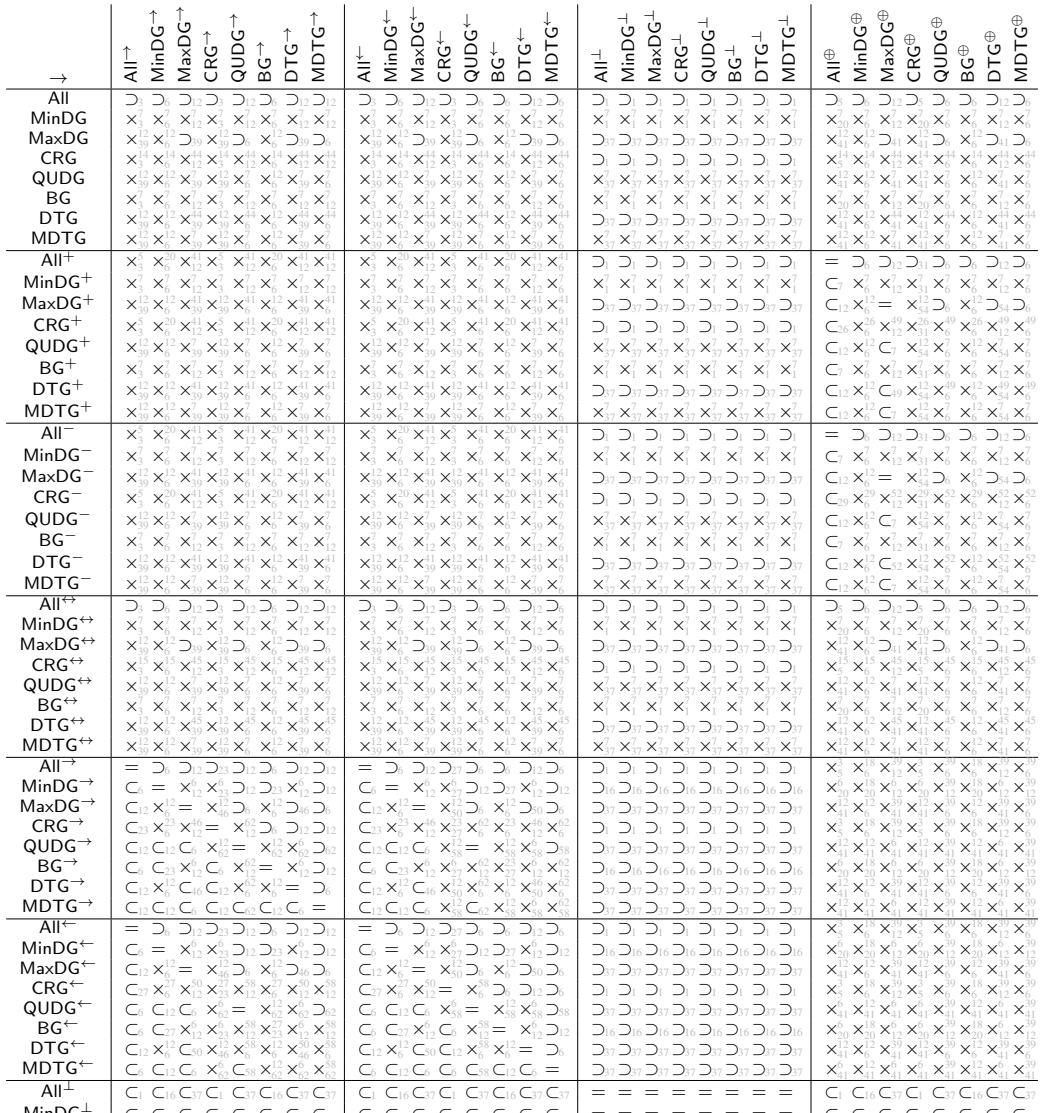
9 Appendix

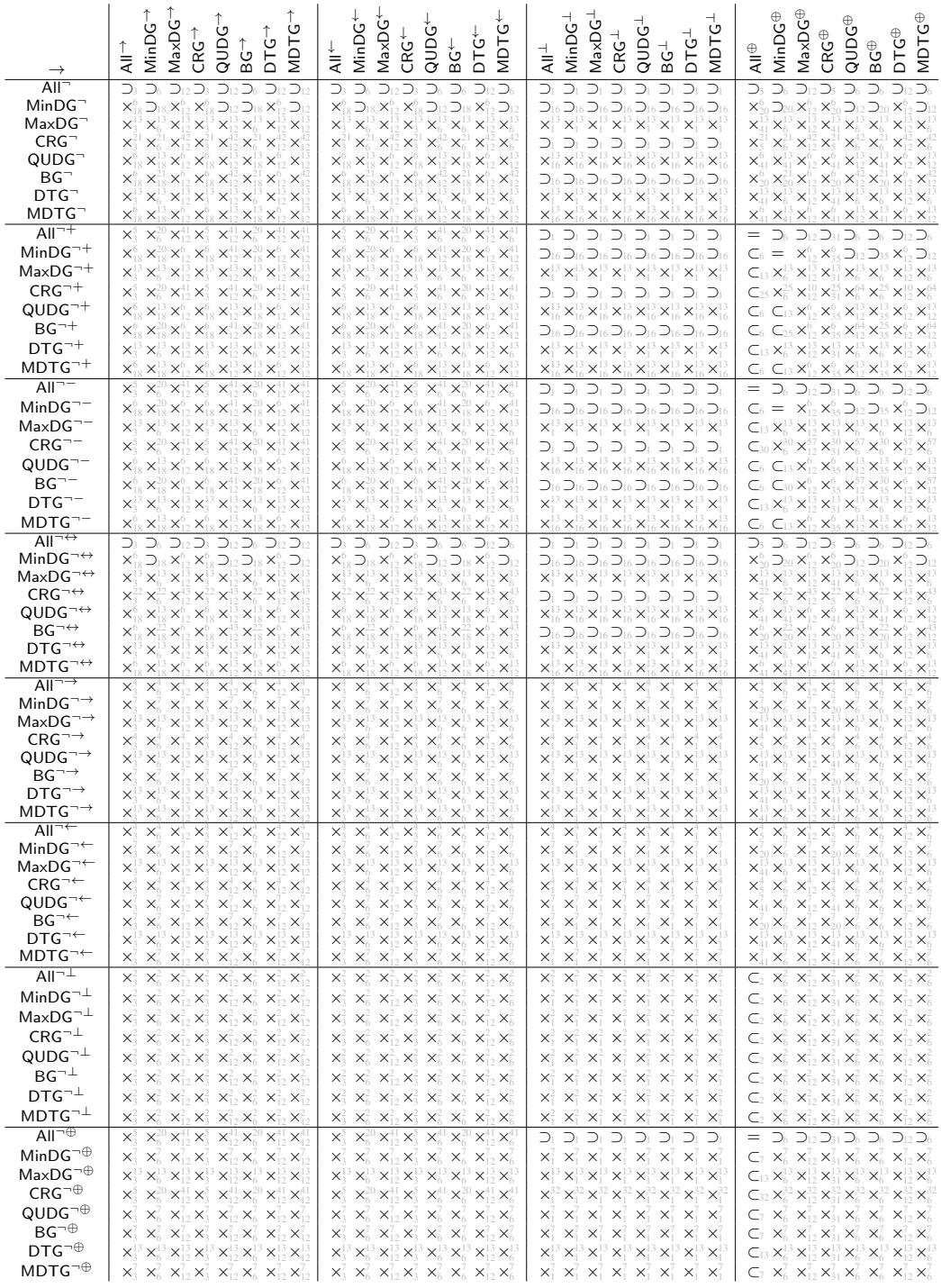
9.1 Automatically Derived Containedness Relations

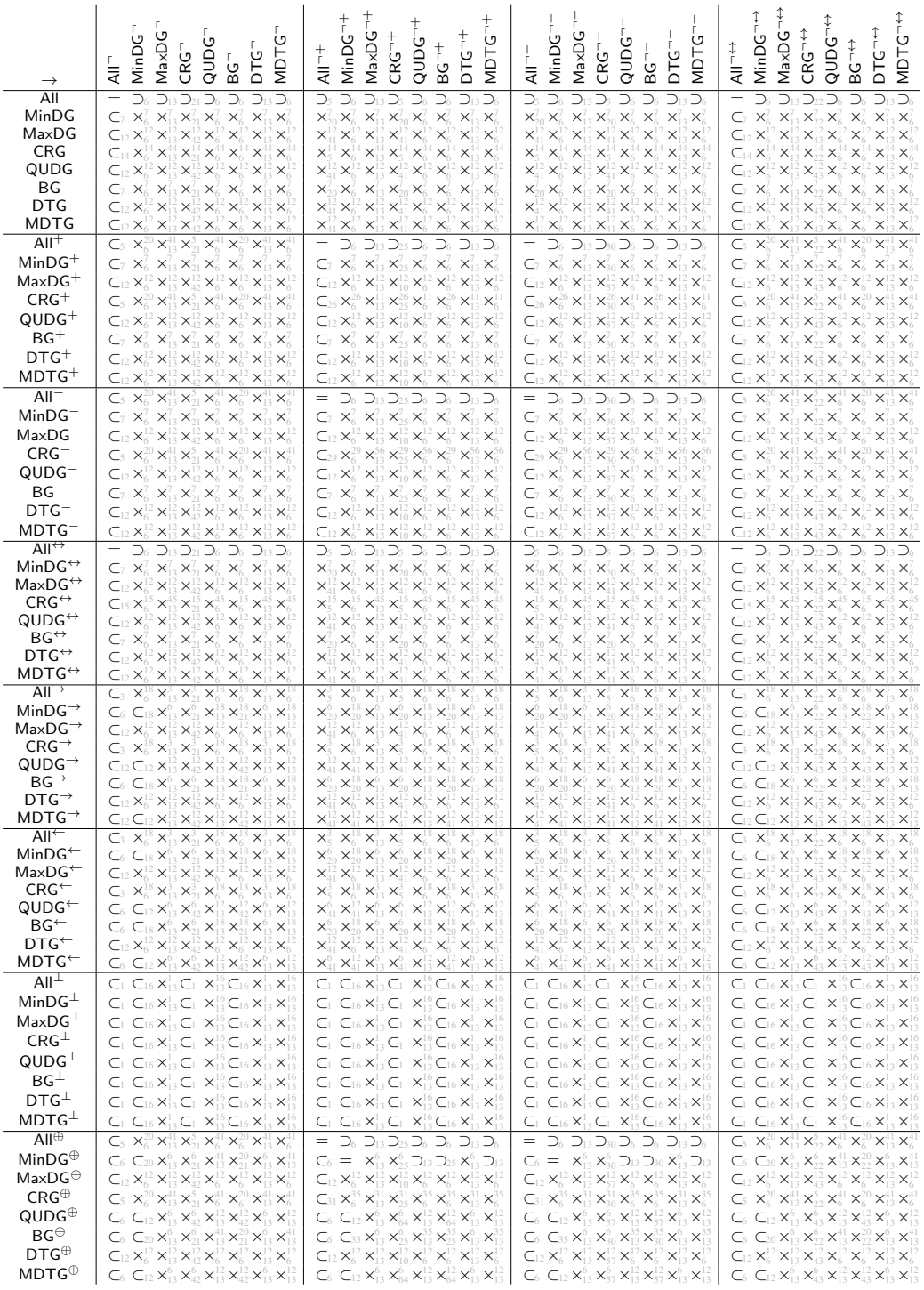
The following tables show the results of the containedness relations tests by Mathematica from Section 4.4.2

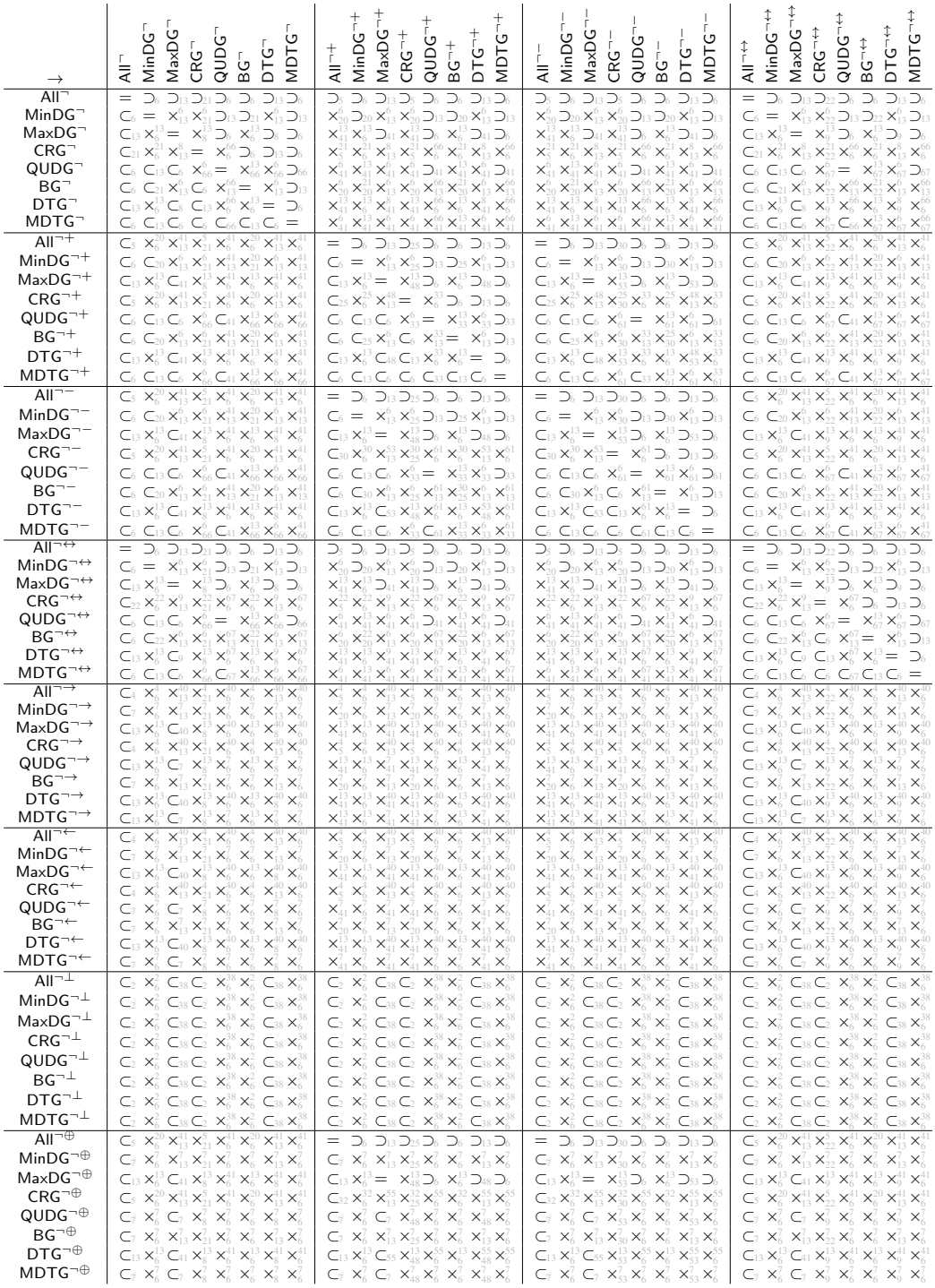


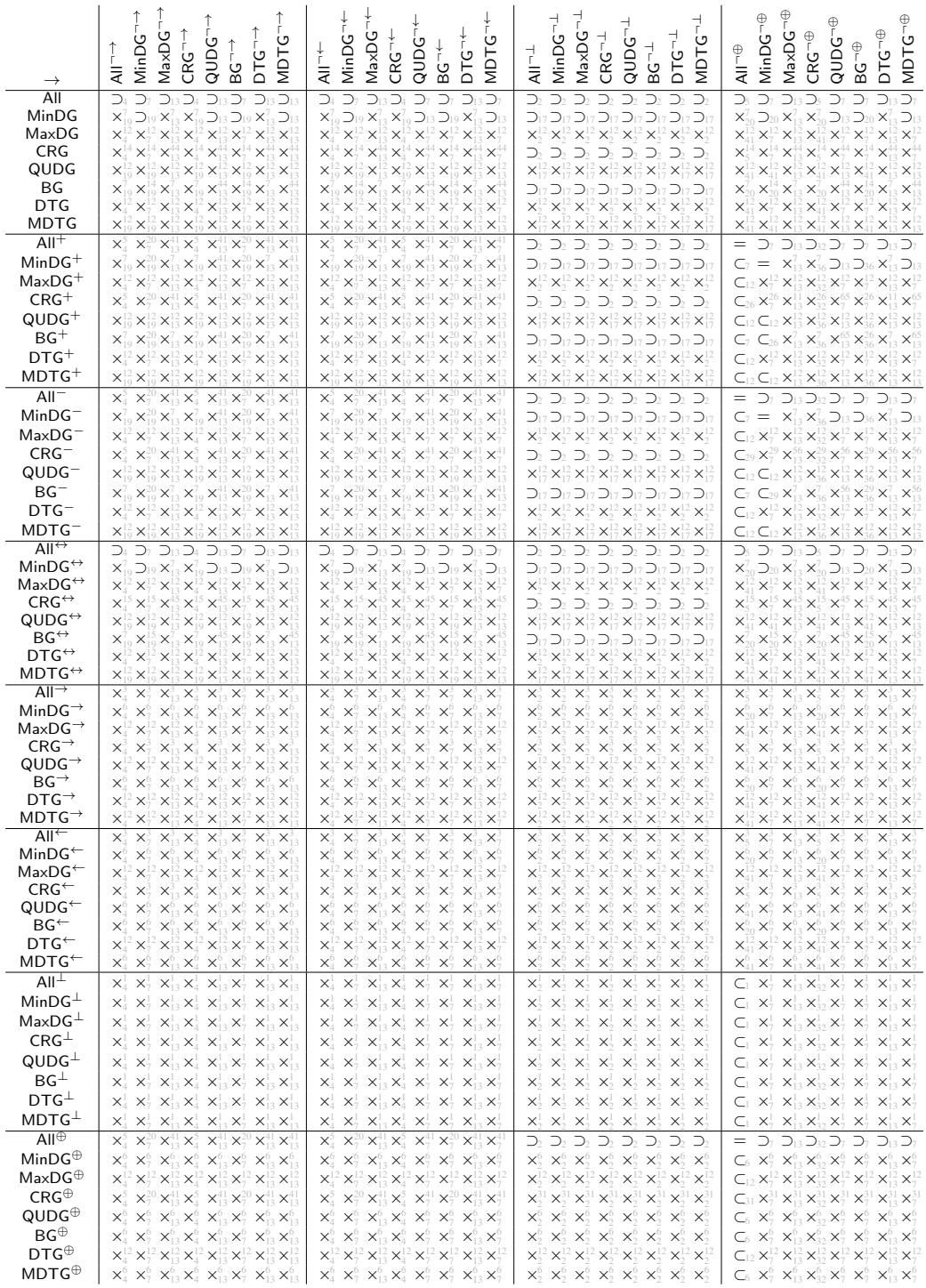

→
All
MinDG
MaxDG
CRG
QUDG
BG
DTG
MDTG
All+
MinDG+
MaxDG+
CRG+
QUDG+
BG+
DTG+
MDTG+
All−
MinDG−
MaxDG−
CRG−
QUDG−
BG−
DTG−
MDTG−
All↔
MinDG↔
MaxDG↔
CRG↔
QUDG↔
BG↔
DTG↔
MDTG↔
All¬
MinDG¬
MaxDG¬
CRG¬
QUDG¬
BG¬
DTG¬
MDTG¬
All¬+
MinDG¬+
MaxDG¬+
CRG¬+
QUDG¬+
BG¬+
DTG¬+
MDTG¬+
All¬−
MinDG¬−
MaxDG¬−
CRG¬−
QUDG¬−
BG¬−
DTG¬−
MDTG¬−
All¬↔
MinDG¬↔
MaxDG¬↔
CRG¬↔
QUDG¬↔
BG¬↔
DTG¬↔
MDTG¬↔
All¬→
MinDG¬→
MaxDG¬→
CRG¬→
QUDG¬→
BG¬→
DTG¬→
MDTG¬→
All¬←
MinDG¬←
MaxDG¬←
CRG¬←
QUDG¬←
BG¬←
DTG¬←
MDTG¬←
All¬⊥
MinDG¬⊥
MaxDG¬⊥
CRG¬⊥
QUDG¬⊥
BG¬⊥
DTG¬⊥
MDTG¬⊥
All¬⊕
MinDG¬⊕
MaxDG¬⊕
CRG¬⊕
QUDG¬⊕
BG¬⊕
DTG¬⊕
MDTG¬⊕

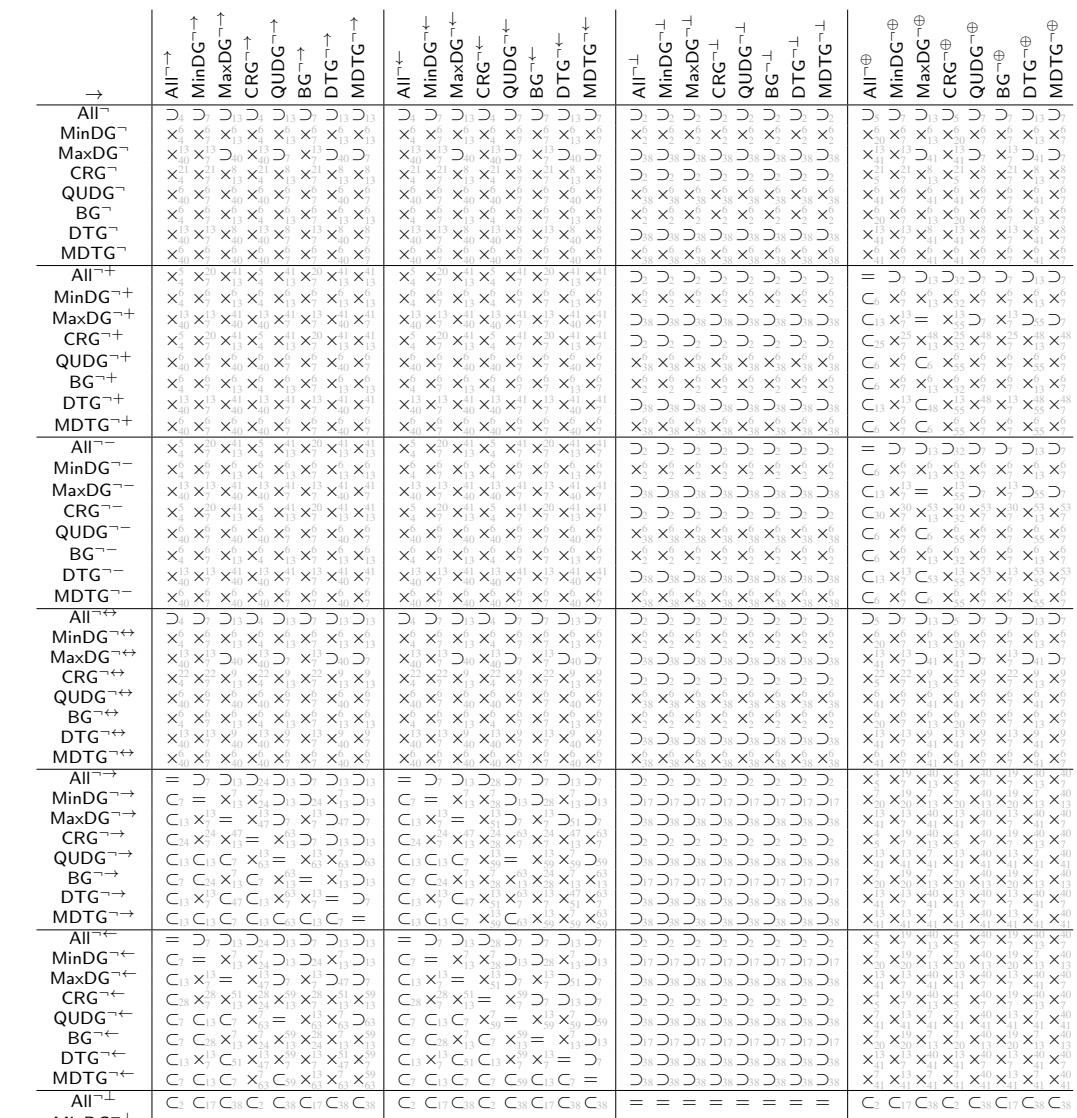






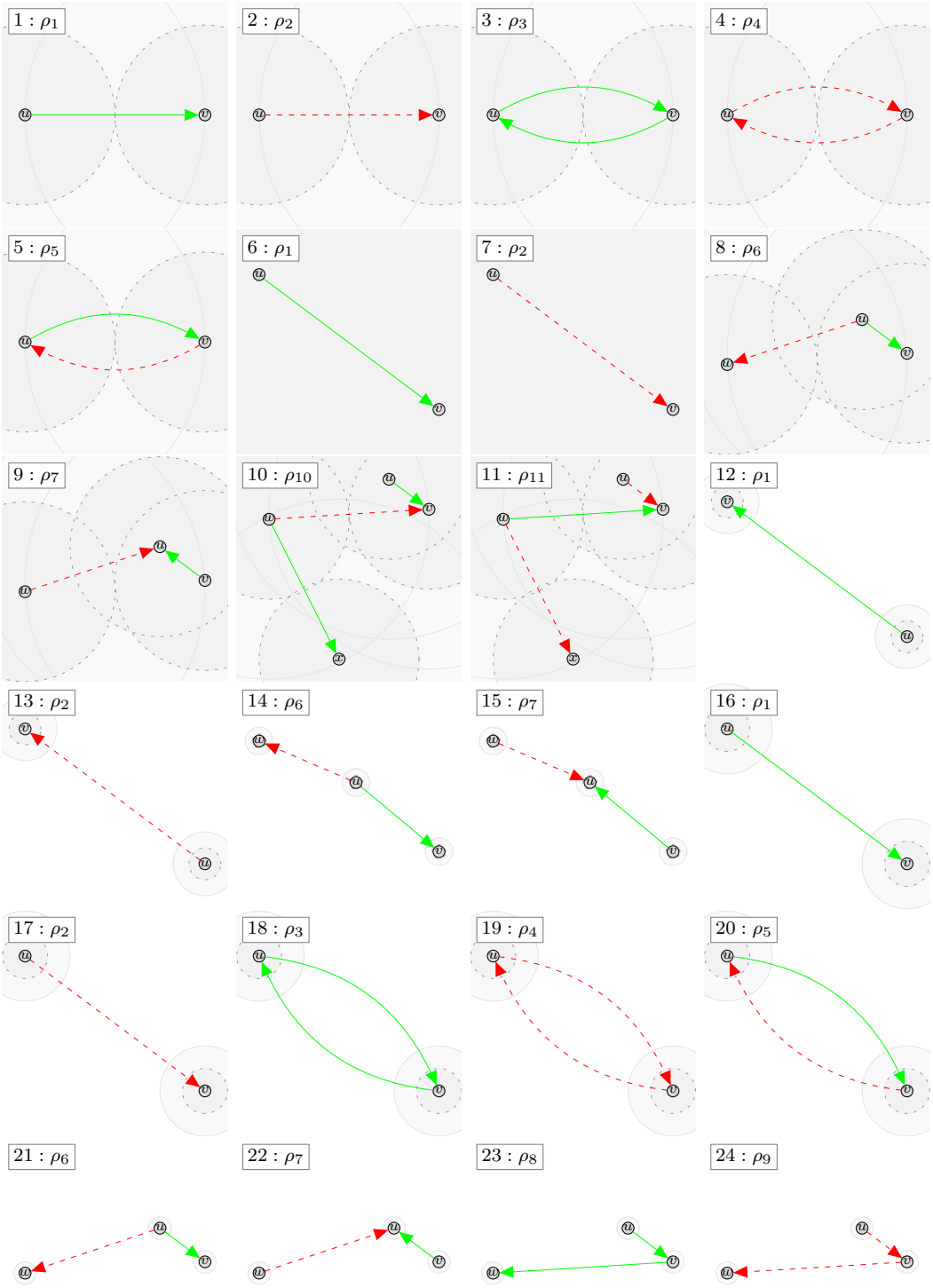

→
All¬
MinDG¬
MaxDG¬
CRG¬
QUDG¬
BG¬
DTG¬
MDTG¬
All¬+
MinDG¬+
MaxDG¬+
CRG¬+
QUDG¬+
BG¬+
DTG¬+
MDTG¬+
All¬−
MinDG¬−
MaxDG¬−
CRG¬−
QUDG¬−
BG¬−
DTG¬−
MDTG¬−
All¬↔
MinDG¬↔
MaxDG¬↔
CRG¬↔
QUDG¬↔
BG¬↔
DTG¬↔
MDTG¬↔
All¬→
MinDG¬→
MaxDG¬→
CRG¬→
QUDG¬→
BG¬→
DTG¬→
MDTG¬→
All¬←
MinDG¬←
MaxDG¬←
CRG¬←
QUDG¬←
BG¬←
DTG¬←
MDTG¬←
All¬⊥
MinDG¬⊥
MaxDG¬⊥
CRG¬⊥
QUDG¬⊥
BG¬⊥
DTG¬⊥
MDTG¬⊥
All¬⊕
MinDG¬⊕
MaxDG¬⊕
CRG¬⊕
QUDG¬⊕
BG¬⊕
DTG¬⊕
MDTG¬⊕

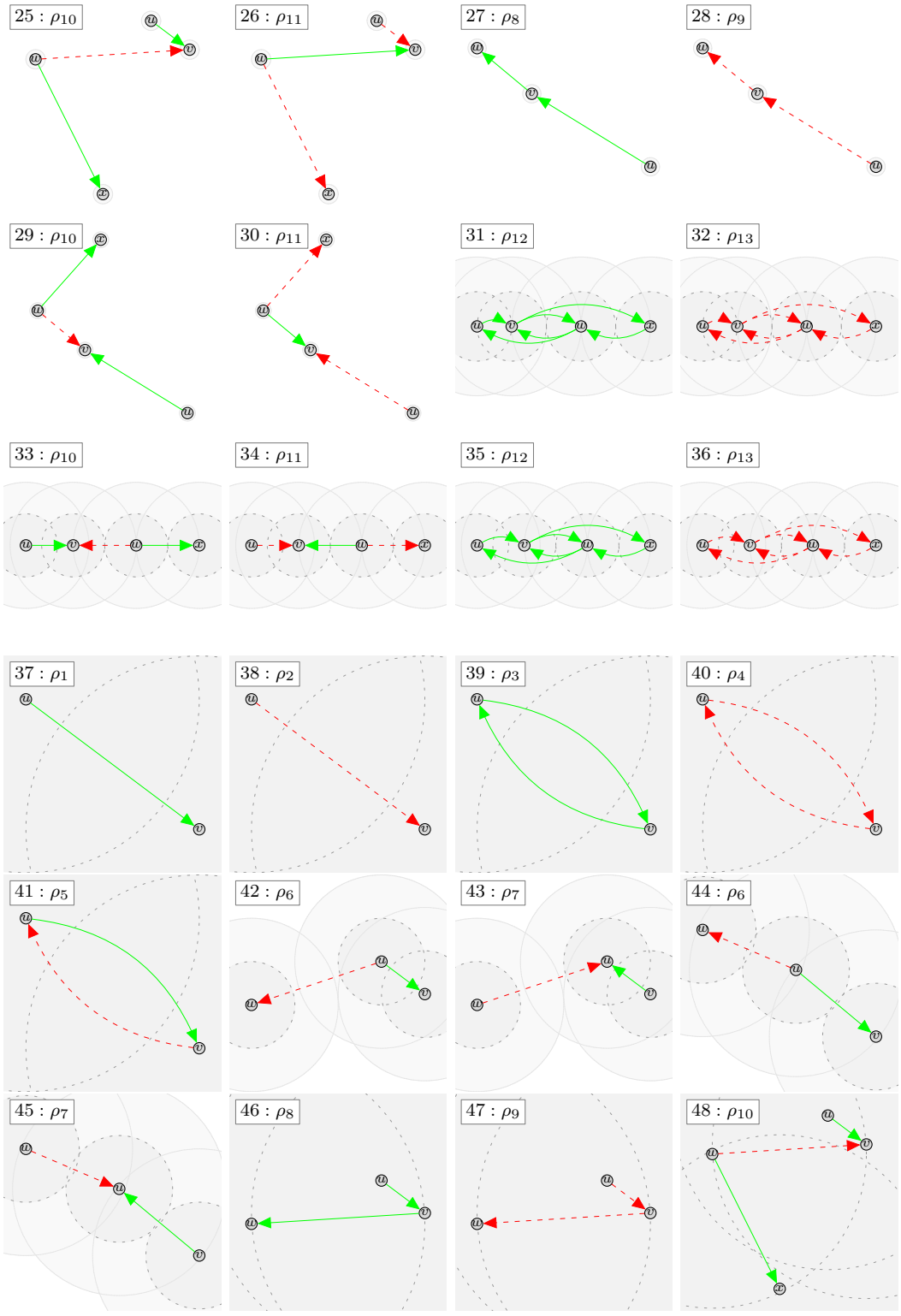


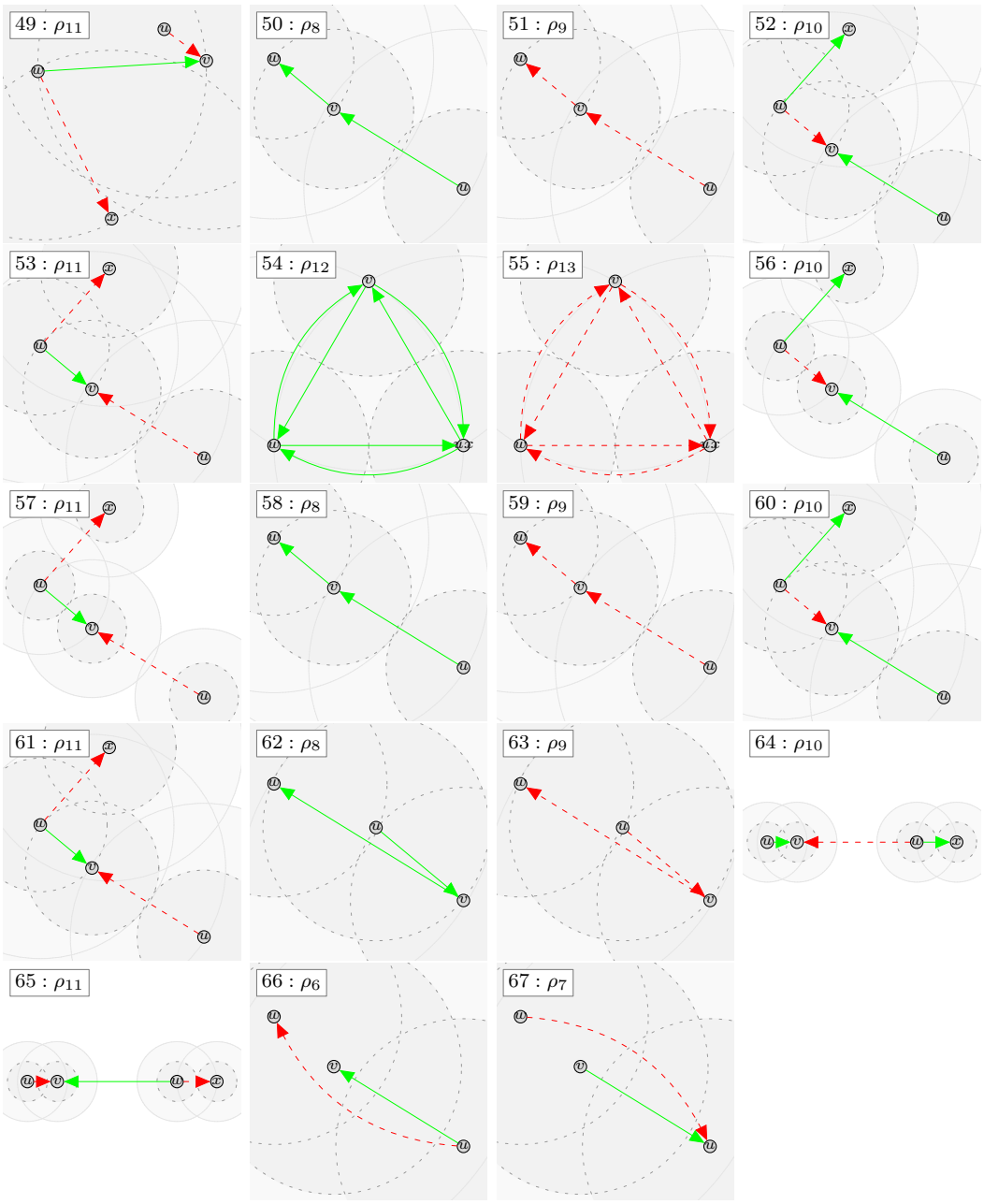


9.2 Automatically Derived Counter Examples

The following graph patterns correspond to the counterexamples obtained in the containedness relation tests by Mathematica in Section 4.4.2.




25 : ρ_{10}
26 : ρ_{11}
27 : ρ_{8}
28 : ρ_{9}
29 : ρ_{10}
30 : ρ_{11}
31 : ρ_{12}
32 : ρ_{13}
33 : ρ_{10}
34 : ρ_{11}
35 : ρ_{12}
36 : ρ_{13}
37 : ρ_{1}
38 : ρ_{2}
39 : ρ_{3}
40 : ρ_{4}
41 : ρ_{5}
42 : ρ_{6}
43 : ρ_{7}
44 : ρ_{6}
45 : ρ_{7}
46 : ρ_{8}
47 : ρ_{9}
48 : ρ_{10}


49 : ρ_{11}
50 : ρ_8
51 : ρ_9
52 : ρ_{10}
53 : ρ_{11}
54 : ρ_{12}
55 : ρ_{13}
56 : ρ_{10}
57 : ρ_{11}
58 : ρ_8
59 : ρ_9
60 : ρ_{10}
61 : ρ_{11}
62 : ρ_8
63 : ρ_9
64 : ρ_{10}
65 : ρ_{11}
66 : ρ_6
67 : ρ_7

9.3 Overview about the axioms used for the verification of the CP-algorithm

In the following, the axioms of the axiom system used for the proof tasks of the correctness proof for the CP-algorithm are listed. Axioms, which are contained in the axiom system are written in boldface. Furthermore some implications, which can be derived from certain axioms of the axioms system are listed in sans serif.

9.3.1 Axioms for left

A1: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \rightarrow \neg\mathsf{left}(w,u,v))$
A2: $(u \neq w \wedge v \neq w \wedge \neg\mathsf{left}(w,u,v) \wedge \neg\mathsf{left}(w,v,u) \rightarrow \mathsf{left}(u,v,w))$
A3: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(u,x,w) \rightarrow \mathsf{left}(u,x,v))$
A4: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(u,x,v) \rightarrow \mathsf{left}(u,x,w))$
A5: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,w,u) \wedge \mathsf{left}(w,u,v) \rightarrow \mathsf{left}(u,v,x) \vee \mathsf{left}(v,w,x) \vee \mathsf{left}(w,u,x))$

(for 5 different points **A5** is implied by **A1-4** and **A6**)

Axiom for a triangle Δuvw left of xy and z inside Δuvw

A6: $\mathsf{left}(u,v,z) \wedge \mathsf{left}(v,w,z) \wedge \mathsf{left}(w,u,z) \wedge \mathsf{left}(x,y,u) \wedge \mathsf{left}(x,y,v) \wedge \mathsf{left}(x,y,w)$
$\rightarrow \mathsf{left}(x,y,z)$

Axioms implied by A1-3

B1: $\neg\mathsf{left}(u,u,v)$
B2: $\neg\mathsf{left}(u,v,u)$
B3: $(u \neq v \rightarrow \mathsf{left}(u,v,v))$

B11: $(u \neq v \rightarrow \mathsf{left}(u,v,w) \vee \mathsf{left}(v,u,w))$
B12: $(u \neq w \rightarrow \mathsf{left}(u,v,w) \vee \mathsf{left}(w,v,u))$

B21: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \rightarrow \mathsf{left}(u,w,v))$
B22: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \rightarrow \mathsf{left}(v,w,u))$
B23: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \rightarrow \neg\mathsf{left}(w,u,v))$
B24: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \rightarrow \neg\mathsf{left}(w,v,u))$

B31: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(w,v,u) \rightarrow \mathsf{left}(u,w,v))$
B32: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(w,v,u) \rightarrow \mathsf{left}(w,u,v))$
B33: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(w,v,u) \rightarrow \neg\mathsf{left}(v,u,w))$
B34: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(w,v,u) \rightarrow \neg\mathsf{left}(v,w,u))$

Axioms implied by A1-5

B41: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(u,w,v) \rightarrow \mathsf{left}(v,u,w) \vee \mathsf{left}(w,v,u) \vee v = w)$
B42: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(u,w,v) \rightarrow \mathsf{left}(v,w,u) \vee \mathsf{left}(w,u,v) \vee v = w)$

B51: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(w,u,v) \rightarrow \neg\mathsf{left}(v,u,w))$

B52: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(w,u,v) \rightarrow \mathsf{left}(u,w,v) \vee \mathsf{left}(v,w,u))$
B53: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(u,w,v) \rightarrow \mathsf{left}(w,v,u) \vee \mathsf{left}(v,w,u))$

B61: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,w,u) \rightarrow \neg\mathsf{left}(w,v,u))$
B62: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,w,u) \rightarrow \mathsf{left}(v,u,w) \vee \mathsf{left}(w,u,v))$
B63: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,w,u) \rightarrow \mathsf{left}(u,w,v) \vee \mathsf{left}(w,u,v))$

Axioms implied by A1-6

B71: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(w,u,x) \rightarrow \mathsf{left}(v,u,x))$
B72: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(w,x,u) \rightarrow \mathsf{left}(v,x,u))$
B73: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(x,v,u) \rightarrow \mathsf{left}(x,v,w))$
B74: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(x,v,u) \rightarrow \mathsf{left}(x,w,u))$
B75: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(u,w,x) \rightarrow \mathsf{left}(u,v,x))$
B76: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(u,v,x) \rightarrow \mathsf{left}(u,w,x))$
B77: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(w,u,x) \rightarrow \neg\mathsf{left}(w,v,x))$
B78: $(\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(w,x,u) \rightarrow \neg\mathsf{left}(w,x,v))$

9.3.2 Axioms for intersection

intersection $\rightarrow$

I1-2: $\mathsf{intersection}(u,v,w,x) \rightarrow (u \neq w \vee v \neq x) \wedge (u \neq x \vee v \neq w)$
I3: $\mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{left}(u,v,w) \vee \mathsf{left}(u,v,x)$
I4: $\mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{left}(v,u,w) \vee \mathsf{left}(v,u,x)$
I5: $\mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{left}(w,x,u) \vee \mathsf{left}(w,x,v)$
I6: $\mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{left}(x,w,u) \vee \mathsf{left}(x,w,v)$

intersection $\leftarrow$

I11-12: $\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,x) \wedge \mathsf{left}(w,x,u) \wedge \mathsf{left}(x,w,v)$
$\rightarrow \mathsf{intersection}(u,v,w,x) \vee (u = x \wedge v = w)$
I13: $\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,x) \wedge \mathsf{left}(w,x,v) \wedge \mathsf{left}(x,w,u) \rightarrow \mathsf{intersection}(u,v,w,x)$
I14: $\mathsf{left}(u,v,x) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(w,x,u) \wedge \mathsf{left}(x,w,v) \rightarrow \mathsf{intersection}(u,v,w,x)$
I15-16: $\mathsf{left}(u,v,x) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{left}(w,x,v) \wedge \mathsf{left}(x,w,u)$
$\rightarrow \mathsf{intersection}(u,v,w,x) \vee (u = w \wedge v = x)$

9.3.3 Axioms for inside

Axioms defining inside

T1-T3: $\mathsf{inside}(u,v,w,x) \rightarrow \mathsf{left}(u,v,x) \wedge \mathsf{left}(v,w,x) \wedge \mathsf{left}(w,u,x)$

inside $\rightarrow$

T4: $\mathsf{left}(u,v,x) \wedge \mathsf{left}(v,w,x) \wedge \mathsf{left}(w,u,x) \rightarrow \mathsf{inside}(u,v,w,x)$
T5: $\mathsf{left}(u,v,x) \wedge \mathsf{left}(v,w,x) \wedge \mathsf{left}(w,u,x) \rightarrow \mathsf{inside}(v,w,u,x)$
T6: $\mathsf{left}(u,v,x) \wedge \mathsf{left}(v,w,x) \wedge \mathsf{left}(w,u,x) \rightarrow \mathsf{inside}(w,u,v,x)$

transitivity of inside derived from the other axioms

T11-13: $(\mathsf{inside}(u,v,x,y) \vee \mathsf{inside}(v,w,x,y) \vee \mathsf{inside}(w,u,x,y)) \wedge \mathsf{inside}(u,v,w,x)$
$\rightarrow \mathsf{inside}(u,v,w,y)$
T14: $\mathsf{inside}(u,v,w,y) \wedge \mathsf{inside}(u,v,w,x) \wedge x \neq y$
$\rightarrow (\mathsf{inside}(u,v,x,y) \vee \mathsf{inside}(v,w,x,y) \vee \mathsf{inside}(w,u,x,y))$

9.3.4 Axioms for edges E in the input graph of the algorithm

E1: $\neg\mathsf{E}(u,u)$ (no self-loops)
E2: $\mathsf{E}(u,v) \rightarrow \mathsf{E}(v,u)$ (symmetry of edges)
E3: $\mathsf{E}(u,v) \rightarrow \mathsf{V}(u)$ (symmetry implies that also v is a vertex)

R1: $\mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \rightarrow (\mathsf{E}(u,w) \vee \mathsf{E}(v,x))$
R2: $\mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \rightarrow (\mathsf{E}(u,x) \vee \mathsf{E}(v,w))$
(symmetry implies R2)
C1: $\mathsf{intersection}(u,v,w,x) \wedge \mathsf{E}(u,v) \wedge \mathsf{E}(v,w) \wedge \mathsf{E}(w,u) \wedge \mathsf{V}(x) \rightarrow \mathsf{E}(u,x)$
C2: $\mathsf{intersection}(u,v,w,x) \wedge \mathsf{E}(u,v) \wedge \mathsf{E}(v,w) \wedge \mathsf{E}(w,u) \wedge \mathsf{V}(x) \rightarrow \mathsf{E}(v,x)$
C3: $\mathsf{intersection}(u,v,w,x) \wedge \mathsf{E}(u,v) \wedge \mathsf{E}(v,w) \wedge \mathsf{E}(w,u) \wedge \mathsf{V}(x) \rightarrow \mathsf{E}(w,x)$
(symmetry implies C2 and C3)

9.3.5 Axioms for edges F in the output graph of the algorithm

Axioms for the definition of F

F1: $\mathsf{F}(u,v) \rightarrow \mathsf{F}(v,u)$
F2: $\mathsf{F}(u,v) \rightarrow \mathsf{E}(u,v)$
F3: $\mathsf{F}(u,v) \wedge \mathsf{F}(w,x) \rightarrow \neg\mathsf{intersection}(u,v,w,x)$
F4: $\mathsf{left}(u,v,w) \wedge \mathsf{left}(v,u,w) \wedge \mathsf{V}(w) \rightarrow \neg\mathsf{F}(u,v)$
(no vertex can be located on an edge in F)
F5: $\mathsf{F}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{E}(u,w) \vee \mathsf{E}(v,w)$(CP-condition)
F6: $\mathsf{F}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \rightarrow \mathsf{E}(u,x) \vee \mathsf{E}(v,x)$(CP-condition)
(F5 and F6 are symmetric)

deleting $\rightarrow$

D1-3: $\mathsf{deleting}(u,v,w,x) \rightarrow \mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x)$
D4-5: $\mathsf{deleting}(u,v,w,x) \rightarrow \mathsf{E}(u,w) \wedge \mathsf{E}(v,w)$
D6: $\mathsf{deleting}(u,v,w,x) \rightarrow \neg\mathsf{F}(u,v)$

E,deleting $\rightarrow$

D11: $\mathsf{E}(u,x) \wedge \mathsf{deleting}(u,v,w,x) \rightarrow \mathsf{F}(w,x)$
D12: $\mathsf{E}(v,x) \wedge \mathsf{deleting}(u,v,w,x) \rightarrow \mathsf{F}(w,x)$

deleting $\leftarrow$

D13: $\mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x)$
$\rightarrow \mathsf{deleting}(u,v,w,x) \vee \mathsf{E}(u,x) \vee \mathsf{E}(v,x)$
D14: $\mathsf{E}(u,v) \wedge \mathsf{F}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \wedge \mathsf{E}(u,w) \wedge \mathsf{E}(v,w) \rightarrow \mathsf{deleting}(u,v,w,x)$
implications derived by the other axioms
D23: $\mathsf{E}(u,v) \wedge \mathsf{E}(w,x) \wedge \mathsf{intersection}(u,v,w,x)$
$\rightarrow \mathsf{deleting}(u,v,x,w) \vee \mathsf{E}(u,w) \vee \mathsf{E}(v,w)$
(equivalent to D13 by replacing w and x)
D24: $\mathsf{E}(u,v) \wedge \mathsf{F}(w,x) \wedge \mathsf{intersection}(u,v,w,x) \wedge \mathsf{E}(u,x) \wedge \mathsf{E}(v,x) \rightarrow \mathsf{deleting}(u,v,x,w)$
(equivalent to D14 by replacing w and x)

9.3.6 Axioms for the convex hull of the vertices in $\Delta u_1 p_1 w_1$

Below, u_1, v_1, w_1 and x_1 are considered to be Skolem constants. p_1 is the intersection of the segments u_1v_1 and w_1x_1.

X1: $\forall y,z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(u_1,v_1,y,z) \rightarrow \neg\mathsf{intersection}(u_1,w_1,y,z))$
X2: $\forall y,z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(u_1,v_1,y,z) \rightarrow \neg\mathsf{intersection}(u_1,x_1,y,z))$
X3: $\forall y,z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(u_1,v_1,y,z) \rightarrow \neg\mathsf{inside}(u_1,x_1,w_1,y))$
X4: $\forall y,z\ (\mathsf{F}(y,z) \wedge \mathsf{intersection}(u_1,v_1,y,z) \rightarrow \neg\mathsf{inside}(u_1,x_1,w_1,z))$

Describing the properties of the convex hull. The vertices i and $\mathsf{next}(i)$ are vertices on the finest path of the convex hull of the triangle $\Delta u_1 p_1 w_1$. The finest path is a path, such that there is no other vertex between i and its successor $\mathsf{next}(i)$. Every vertex y different from i and $\mathsf{next}(i)$ in the area left of u_1v_1 and left of x_1w_1 is left of i and $\mathsf{next}(i)$ or i is located between y and $\mathsf{next}(i)$

Y0: $\mathsf{next}(w_1) = \mathsf{nil}$
Y1: $\forall y,i\ (\mathsf{V}(y) \wedge \mathsf{left}(u_1,v_1,y) \wedge \mathsf{left}(x_1,w_1,y) \rightarrow i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee$
$y = i \vee y = \mathsf{next}(i) \vee \mathsf{left}(i,\mathsf{next}(i),y) \vee \mathsf{left}(y,\mathsf{next}(i),i))$
Y2: $\forall y,i\ (\mathsf{V}(y) \wedge \mathsf{left}(u_1,v_1,y) \wedge \mathsf{left}(x_1,w_1,y) \rightarrow i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee$
$y = i \vee y = \mathsf{next}(i) \vee \mathsf{left}(i,\mathsf{next}(i),y) \vee \mathsf{left}(\mathsf{next}(i),y,i))$
Y3: $\forall y,i\ (\mathsf{V}(y) \wedge \mathsf{left}(u_1,v_1,y) \wedge \mathsf{left}(x_1,w_1,y) \wedge \mathsf{left}(\mathsf{next}(i),i,y) \rightarrow$
$i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee y = i \vee y = \mathsf{next}(i) \vee \mathsf{left}(y,\mathsf{next}(i),i))$
Y4: $\forall y,i\ (\mathsf{V}(y) \wedge \mathsf{left}(u_1,v_1,y) \wedge \mathsf{left}(x_1,w_1,y) \wedge \mathsf{left}(\mathsf{next}(i),i,y) \rightarrow$
$i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee y = i \vee y = \mathsf{next}(i) \vee \mathsf{left}(\mathsf{next}(i),y,i))$

The vertex $\mathsf{next}(i)$ is located inside the triangle $\Delta u_1 v_1 w_1$ or $\mathsf{next}(i) = w_1$. From the other axioms it follows that $\mathsf{next}(i)$ is also inside the triangle $\Delta u_1 x_1 w_1$ and therefore inside the triangle $\Delta u_1 p_1 w_1$ or equal to w_1.

Y11: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(u_1, v_1, w_1, \mathsf{next}(i)) \vee \mathsf{next}(i) = w_1)$

The vertex (i) is located inside the triangle $\Delta u_1 x_1 w_1$ or $i = u_1$. From the other axioms it follows that i is also inside the triangle $\Delta u_1 v_1 w_1$ and therefore inside the triangle $\Delta u_1 p_1 w_1$ or equal to v_1.
Y12: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(u_1, x_1, w_1, i) \vee i = u_1)$
There exists an edge between the vertices of the convex hull.
Y13: $\forall i\ (\mathsf{E}(i, \mathsf{next}(i)) \vee i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil})$

The vertex i is located inside the triangles $\Delta u_1 v_1 \mathsf{next}(i)$ and $\Delta u_1 x_1 \mathsf{next}(i)$ or equal to u_1
Y21: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(u_1, v_1, \mathsf{next}(i), i) \vee i = u_1)$
Y22: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(u_1, x_1, \mathsf{next}(i), i) \vee i = u_1)$
The vertex $\mathsf{next}(i)$ is located inside the triangles $\Delta v_1 w_1 i$ and $\Delta x_1 w_1 i$ or equal to w_1
Y23: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(v_1, w_1, i, \mathsf{next}(i)) \vee \mathsf{next}(i) = w_1)$
Y24: $\forall i\ (i = \mathsf{nil} \vee \mathsf{next}(i) = \mathsf{nil} \vee \mathsf{inside}(x_1, w_1, i, \mathsf{next}(i)) \vee \mathsf{next}(i) = w_1)$

9.4 Proof tasks

In the following, the proof tasks of the correctness of the CP-algorithm are listed. For proof tasks considering a redundancy sequence of eligible edges obtained by deleting edges, the eligible and deleting edges, are listed on the right side.

9.4.1 Proof tasks of closestF

Step1a:

$$
\begin{aligned}
& \mathsf{left}(u_1, v_1, p_1) \wedge \mathsf{left}(v_1, u_1, p_1) \wedge \mathsf{left}(w_1, x_1, p_1) \wedge \mathsf{left}(x_1, w_1, p_1) \\
\wedge\ & \mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge F(w_1, x_1) \\
\wedge\ & \mathsf{deleting}(u_1, w_1, w_2, x_2) \wedge F(w_2, x_2) \\
\rightarrow\ & \mathsf{inside}(u_1, p_1, w_1, w_2) \vee \mathsf{inside}(u_1, p_1, w_1, x_2) \\
\vee\ & \mathsf{inside}(u_1, w_1, p_1, w_2) \vee \mathsf{inside}(u_1, w_1, p_1, x_2) \\
\vee\ & \mathsf{intersection}(u_1, p_1, w_2, x_2)
\end{aligned}
$$

Step1b:

$$
\begin{aligned}
& \mathsf{left}(u_1, v_1, p_1) \wedge \mathsf{left}(v_1, u_1, p_1) \wedge \mathsf{left}(w_1, x_1, p_1) \wedge \mathsf{left}(x_1, w_1, p_1) \\
\wedge\ & \mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge F(w_1, x_1) \\
\wedge\ & \mathsf{deleting}(u_1, w_1, w_2, x_2) \wedge F(w_2, x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, w_2, x_2) \\
\rightarrow\ & \mathsf{intersection}(u_1, p_1, w_2, x_2)
\end{aligned}
$$

Step1c:

$\mathsf{left}(u_1, v_1, p_1) \wedge \mathsf{left}(v_1, u_1, p_1) \wedge \mathsf{left}(w_1, x_1, p_1) \wedge \mathsf{left}(x_1, w_1, p_1)$
$\wedge\ \mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge F(w_1, x_1)$
$\wedge\ \mathsf{deleting}(u_1, w_1, x_1, x_2) \wedge F(x_1, x_2)$
$\rightarrow\ \mathsf{intersection}(u_1, p_1, x_1, x_2)$

Step2a:

$\mathsf{left}(u_1, v_1, p_1) \wedge \mathsf{left}(v_1, u_1, p_1) \wedge \mathsf{left}(w_1, x_1, p_1) \wedge \mathsf{left}(x_1, w_1, p_1)$
$\wedge\ \mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge F(w_1, x_1)$
$\wedge\ \mathsf{intersection}(u_1, p_1, w_2, x_2) \wedge F(w_2, x_2)$
$\rightarrow\ \mathsf{inside}(u_1, p_1, w_1, w_2) \vee \mathsf{inside}(u_1, p_1, w_1, x_2)$
$\vee\ \mathsf{inside}(u_1, w_1, p_1, w_2) \vee \mathsf{inside}(u_1, w_1, p_1, x_2)$
$\vee\ \mathsf{intersection}(u_1, w_1, w_2, x_2)$
$\vee\ w_2 = w_1 \vee x_2 = w_1$

Step2b:

$\mathsf{left}(u_1, v_1, p_1) \wedge \mathsf{left}(v_1, u_1, p_1) \wedge \mathsf{left}(w_1, x_1, p_1) \wedge \mathsf{left}(x_1, w_1, p_1)$
$\wedge\ \mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge F(w_1, x_1)$
$\wedge\ \mathsf{intersection}(u_1, p_1, w_1, x_2) \wedge F(w_1, x_2)$
$\rightarrow\ \mathsf{inside}(u_1, p_1, x_1, x_2) \vee \mathsf{inside}(u_1, x_1, p_1, x_2)$
$\vee\ \mathsf{intersection}(u_1, x_1, w_1, x_2)$
$\vee\ x_2 = x_1$

9.4.2 Proof tasks of orderF

Step1:

$\mathsf{deleting}(w_1, x_1, w_2, x_2)$ $(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge\ \mathsf{deleting}(w_1, w_2, w_3, x_3)$ $(e_2 = w_1w_2, d_3 = w_3x_3)$
$\wedge\ \mathsf{deleting}(w_1, x_1, x_3, w_3)$ $(d_{2*} = x_3w_3)$
$\rightarrow\ \mathsf{deleting}(w_1, x_1, w_3, x_3)$ $(d'_2 = w_3x_3)$
$\vee\ \mathsf{deleting}(w_1, w_2, x_3, w_3)$ $(d_{3b} = x_3w_3(Part2\ Step2))$
$\vee\ \mathsf{F}(w_2, x_2)$

Step2:

	$\mathsf{deleting}(w_1, x_1, w_2, x_2)$	$(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge$	$\mathsf{deleting}(w_1, w_2, w_3, x_3)$	$(e_2 = w_1w_2, d_3 = w_3x_3)$
$\wedge$	$\mathsf{deleting}(w_1, w_3, w_4, x_4)$	$(e_3 = w_1w_3, d_4 = w_4x_4)$
$\wedge$	$\mathsf{deleting}(w_1, x_1, x_4, w_4)$	$(d_{2*} = x_4w_4)$
$\rightarrow$	$\mathsf{deleting}(w_1, x_1, w_3, x_3)$	$(d'_2 = w_3x_3)$
$\vee$	$\mathsf{deleting}(w_1, w_2, x_3, w_3)$	$(d_{3b} = x_3w_3(Part2\ Step2))$
$\vee$	$\mathsf{deleting}(w_1, x_1, w_4, x_4)$	$(d'_2 = w_4x_4)$
$\vee$	$\mathsf{deleting}(w_1, w_3, x_4, w_4)$	$(d_{4b} = x_4w_4)(Part2\ Step2))$
$\vee$	$\mathsf{F}(w_2, x_2)$	

9.4.3 Proof tasks of Part1

It is proven that $\mathsf{AxGeom} \cup \mathsf{AxGraphs} \cup \{\mathbf{R1}, \mathbf{C1}\}$ entails:
Step1a:

	$\mathsf{deleting}(u_1, v_1, w_1, x_1)$	$(e_0 = u_1v_1, e_1 = d_1 = w_1x_1)$
$\wedge$	$\mathsf{deleting}(w_1, x_1, w_2, x_2)$	$(d_2 = w_2x_2)$
$\rightarrow$	$\mathsf{deleting}(u_1, v_1, w_2, x_1)$	$(e_2 = d'_1 = w_2x_1(Step1))$
$\vee$	$\mathsf{intersection}(u_1, v_1, w_1, w_2)$	$(e_2 = w_1w_2(Step2))$

Step1b1:

	$\mathsf{deleting}(u_1, v_1, w_1, x_1)$	$(e_0 = u_1v_1, e_1 = d_1 = w_1x_1)$
$\wedge$	$\mathsf{deleting}(w_1, x_1, w_2, x_2)$	$(d_2 = w_2x_2)$
$\wedge$	$\mathsf{inside}(w_1, x_1, w_2, u_1)$	
$\rightarrow$	$\perp$	

Step1b2:

	$\mathsf{deleting}(u_1, v_1, w_1, x_1)$	$(e_0 = u_1v_1, e_1 = d_1 = w_1x_1)$
$\wedge$	$\mathsf{deleting}(w_1, x_1, w_2, x_2)$	$(d_2 = w_2x_2)$
$\wedge$	$\mathsf{inside}(x_1, w_1, w_2, u_1)$	
$\rightarrow$	$\perp$	

Step2a1:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	deleting(y_2, z_1, w_2, x_2)	$(d_2(Z) = w_2x_2)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	deleting(y_2, w_2, w_3, x_3)	$(d_3 = w_3x_3)$
$\wedge\neg$	deleting(y_2, z_1, x_3, w_3)	$(\leftarrow orderF)$
$\rightarrow$	intersection(u_1, v_1, y_2, w_3)	$(d_3(Z), e_3 = y_2w_3(\rightarrow Step4/6))$
$\vee$	intersection(u_1, v_1, w_3, w_2)	$(d_3(Y), e_3 = w_3w_2(\rightarrow Step4))$
$\vee$	deleting(y_2, w_2, x_3, w_3)	$(d_{3b} = x_3w_3(\rightarrow Step3))$

Step2a2:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	deleting(y_2, z_1, w_2, x_2)	$(d_2(Z) = w_2x_2)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	deleting(y_2, w_2, w_3, x_3)	$(d_3 = w_3x_3)$
$\wedge$	intersection(u_1, v_1, y_2, w_3)	$(e_3 = y_2w_3)$
$\wedge\neg$	deleting(y_2, z_1, x_3, w_3)	$(\leftarrow orderF)$
$\rightarrow$	$E(z_1, w_3)$	
$\vee$	$z_1 = w_3$	$(redundancy\ cycle(\rightarrow Part2))$
$\vee$	deleting(y_2, w_2, x_3, w_3)	$(d_{3b} = x_3w_3(\rightarrow Step3))$

Step2a3:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	deleting(y_2, z_1, w_2, x_2)	$(d_2(Z) = w_2x_2)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	intersection(u_1, v_1, y_2, z_2)	$(e_3 = y_2z_2)$
$\wedge$	inside(w_2, z_1, y_2, z_2)	
$\rightarrow$	$E(z_1, z_2)$	

Step2a4:

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_2) & (e_3 = y_2z_2) \\
\wedge & \mathsf{inside}(z_1, w_2, y_2, z_2) & \\
\rightarrow & E(z_1, z_2) &
\end{array}
$$

Step2a5:

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3 = w_3x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_2) & (e_4 = y_2z_2) \\
\wedge & \mathsf{inside}(w_2, w_3, y_2, z_2) & \\
\wedge\neg & \mathsf{deleting}(y_2, z_1, x_3, w_3) & (\leftarrow orderF) \\
\rightarrow & E(z_1, z_2) & \\
\vee & z_1 = z_2 & (redundancy\ cycle(\rightarrow Part2)) \\
\vee & \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Step3))
\end{array}
$$

Step2a6:

$$
\begin{array}{lll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3 = w_3x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_2) & (e_4 = y_2z_2) \\
\wedge & \mathsf{inside}(w_3, w_2, y_2, z_2) & \\
\wedge\neg & \mathsf{deleting}(y_2, z_1, x_3, w_3) & (\leftarrow orderF) \\
\rightarrow & E(z_1, z_2) & \\
\vee & z_1 = z_2 & (redundancy\ cycle(\rightarrow Part2) \\
\vee & \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Step3))
\end{array}
$$

Step2b1:

$$
\begin{array}{lr}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge \ \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3 = w_3x_3) \\
\wedge \neg \mathsf{deleting}(y_2, z_1, x_3, w_3) & (\leftarrow orderF) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1) \\
\wedge \ \mathsf{inside}(w_3, z_1, y_2, u_1) & \\
\rightarrow \ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Step3))
\end{array}
$$

Step2b2:

$$
\begin{array}{lr}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge \ \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3 = w_3x_3) \\
\wedge \neg \mathsf{deleting}(y_2, z_1, x_3, w_3) & (\leftarrow orderF) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1) \\
\wedge \ \mathsf{inside}(z_1, w_3, y_2, u_1) & \\
\rightarrow \ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3(Step3))
\end{array}
$$

Step2c1:

$$
\begin{array}{ll}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge E(z_1, z_3) \wedge E(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3z_3) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1) \\
\wedge & \mathsf{inside}(w_3, y_2, z_3, u_1) & \\
\rightarrow & \mathsf{deleting}(y_2, z_3, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Step3))
\end{array}
$$

Step2c2:

$$
\begin{array}{ll}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge E(z_1, z_3) \wedge E(y_2, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3z_3) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1) \\
\wedge & \mathsf{inside}(y_2, w_3, z_3, u_1) & \\
\rightarrow & \mathsf{deleting}(y_2, z_3, x_3, w_3) & (d_{3b} = x_3w_3(\rightarrow Step3))
\end{array}
$$

Step3a:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	deleting(y_2, w_2, w_3, x_3)	$(d_{3a} = w_3x_3)$
$\wedge$	deleting(y_2, w_2, x_3, w_3)	$(d_{3b} = x_3w_3)$
$\wedge\neg$	inside(w_2, z_1, y_2, u_1)	$(\leftarrow Step1/2)$
$\wedge\neg$	inside(w_2, z_1, y_2, v_1)	$(\leftarrow Step1/2)$
$\wedge\neg$	inside(z_1, w_2, y_2, u_1)	$(\leftarrow Step1/2)$
$\wedge\neg$	inside(z_1, w_2, y_2, v_1)	$(\leftarrow Step1/2)$
$\rightarrow$	intersection(u_1, v_1, y_2, w_3)	$(d_3(Z) = d_{3a}, e_3 = y_2w_3(\rightarrow Step4/6))$
$\vee$	intersection(u_1, v_1, y_2, x_3)	$(d_3(Z) = d_{3b}, e_3 = y_2x_3(\rightarrow Step4/6))$
$\vee$	intersection(u_1, v_1, w_3, w_2)	$(d_3(Y) = d_{3a}, e_3 = w_3w_2(\rightarrow Step4))$
$\vee$	intersection(u_1, v_1, x_3, w_2)	$(d_3(Y) = d_{3b}, e_3 = x_3w_2(\rightarrow Step4))$

Step3b1:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	deleting(y_2, w_2, w_3, x_3)	$(d_{3a} = w_3x_3)$
$\wedge$	deleting(y_2, w_2, x_3, w_3)	$(d_{3b} = x_3w_3)$
$\wedge$	intersection(u_1, v_1, y_2, w_3)	$(d_3(Z) = d_{3a}, e_3 = y_2w_3)$
$\wedge\neg$	inside(w_2, z_1, y_2, u_1)	$(\leftarrow Step1/2)$
$\wedge\neg$	inside(w_2, z_1, y_2, v_1)	$(\leftarrow Step1/2)$
$\wedge\neg$	inside(z_1, w_2, y_2, u_1)	$(\leftarrow Step1/2)$
$\wedge\neg$	inside(z_1, w_2, y_2, v_1)	$(\leftarrow Step1/2)$
$\wedge$	inside(z_1, w_3, y_2, u_1)	
$\rightarrow$	intersection(u_1, v_1, y_2, x_3)	$(d_3(Z) = d_{3b}, e_3 = y_2x_3(Step3b3))$

Step3b2:

$$
\begin{array}{ll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) \qquad (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) \qquad (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) \qquad (d_{3a} = w_3x_3) \\
\wedge & \mathsf{deleting}(y_2, w_2, x_3, w_3) \qquad (d_{3b} = x_3w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_3) \qquad (d_3(Z) = d_{3a}, e_3 = y_2w_3) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, u_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, v_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, u_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, v_1) \qquad (\leftarrow Step1/2) \\
\wedge & \mathsf{inside}(w_3, z_1, y_2, u_1) \\
\rightarrow & \mathsf{intersection}(u_1, v_1, y_2, x_3) \qquad (d_3(Z) = d_{3b}, e_3 = y_2x_3(Step3b3))
\end{array}
$$

Step3b3:

$$
\begin{array}{ll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) \qquad (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) \qquad (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) \qquad (d_{3a} = w_3x_3) \\
\wedge & \mathsf{deleting}(y_2, w_2, x_3, w_3) \qquad (d_{3b} = x_3w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_3) \qquad (d_3(Z) = d_{3a}, e_3 = y_2z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, x_3) \qquad (d_3(Z) = d_{3b}, e_3 = y_2x_3) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, u_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, v_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, u_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, v_1) \qquad (\leftarrow Step1/2) \\
\wedge & \mathsf{inside}(w_3, z_1, y_2, u_1) \\
\wedge & \mathsf{inside}(x_3, z_1, y_2, u_1) \\
\rightarrow & \bot
\end{array}
$$

Step3b4:

$$
\begin{array}{ll}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge \ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_3) & (d_3(Z) = d_{3a}, e_3 = y_2w_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, x_3) & (d_3(Z) = d_{3b}, e_3 = y_2x_3) \\
\wedge \neg \mathsf{inside}(w_3, z_1, y_2, u_1) & (\leftarrow Step3b3) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \ \mathsf{inside}(x_3, z_1, y_2, u_1) & \\
\wedge \ \mathsf{inside}(z_1, w_3, y_2, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step3b5:

$$
\begin{array}{ll}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge \ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_3) & (d_3(Z) = d_{3a}, e_3 = y_2w_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, x_3) & (d_3(Z) = d_{3b}, e_3 = y_2x_3) \\
\wedge \neg \mathsf{inside}(w_3, z_1, y_2, u_1) & (\leftarrow Step3b3) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \ \mathsf{inside}(x_3, z_1, y_2, u_1) & \\
\wedge \ \mathsf{inside}(w_3, z_1, y_2, v_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step3b6:

$$
\begin{array}{ll}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge & \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_3) & (d_3(Z) = d_{3a}, e_3 = y_2w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, x_3) & (d_3(Z) = d_{3b}, e_3 = y_2x_3) \\
\wedge\neg & \mathsf{inside}(w_3, z_1, y_2, u_1) & (\leftarrow Step3b3) \\
\wedge\neg & \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge\neg & \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge\neg & \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge\neg & \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge & \mathsf{inside}(x_3, z_1, y_2, u_1) & \\
\wedge & \mathsf{inside}(z_1, w_3, y_2, v_1) & \\
\rightarrow & \bot &
\end{array}
$$

Step3c1:

$$
\begin{array}{ll}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge & \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, w_2) & (d_3(Y) = d_{3a}, e_3 = w_3w_2) \\
\wedge\neg & \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge\neg & \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge\neg & \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge\neg & \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge & \mathsf{inside}(w_3, y_2, w_2, u_1) & \\
\rightarrow & \mathsf{intersection}(u_1, v_1, x_3, w_2) & (d_3(Y) = d_{3b}, e_3 = x_3w_2(Step3c3))
\end{array}
$$

Step3c2:

$$
\begin{array}{ll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) \qquad (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) \qquad (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) \qquad (d_{3a} = w_3x_3) \\
\wedge & \mathsf{deleting}(y_2, w_2, x_3, w_3) \qquad (d_{3b} = x_3w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, w_2) \qquad (d_3(Y) = d_{3a}, e_3 = w_3w_2) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, v_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, u_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, v_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, u_1) \qquad (\leftarrow Step1/2) \\
\wedge & \mathsf{inside}(y_2, w_3, w_2, u_1) \\
\rightarrow & \mathsf{intersection}(u_1, v_1, x_3, w_2) \qquad (d_3(Y) = d_{3b}, e_3 = x_3w_2(Step3c3))
\end{array}
$$

Step3c3:

$$
\begin{array}{ll}
 & \mathsf{deleting}(u_1, v_1, y_2, z_1) \qquad (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) \qquad (e_2 = y_2w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) \qquad (d_{3a} = w_3x_3) \\
\wedge & \mathsf{deleting}(y_2, w_2, x_3, w_3) \qquad (d_{3b} = x_3w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, w_2) \qquad (d_3(Y) = d_{3a}, e_3 = w_3w_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, x_3, w_2) \qquad (d_3(Y) = d_{3b}, e_3 = x_3w_2) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, u_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(w_2, z_1, y_2, v_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, u_1) \qquad (\leftarrow Step1/2) \\
\wedge & \neg\mathsf{inside}(z_1, w_2, y_2, v_1) \qquad (\leftarrow Step1/2) \\
\wedge & \mathsf{inside}(w_3, y_2, w_2, u_1) \\
\wedge & \mathsf{inside}(x_3, y_2, w_2, u_1) \\
\rightarrow & \bot
\end{array}
$$

Step3c4:

$$
\begin{array}{ll}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge \ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, w_3, w_2) & (d_3(Y) = d_{3a}, e_3 = y_2z_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, x_3, w_2) & (d_3(Y) = d_{3b}, e_3 = y_2x_3) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(w_3, y_2, w_2, u_1) & (\leftarrow Step3c3) \\
\wedge \ \mathsf{inside}(x_3, y_2, w_2, u_1) & \\
\wedge \ \mathsf{inside}(y_2, w_3, w_2, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step3c5:

$$
\begin{array}{ll}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge \ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, w_3, w_2) & (d_3(Y) = d_{3a}, e_3 = y_2z_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, x_3, w_2) & (d_3(Y) = d_{3b}, e_3 = y_2x_3) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(w_3, y_2, w_2, u_1) & (\leftarrow Step3c3) \\
\wedge \ \mathsf{inside}(x_3, y_2, w_2, u_1) & \\
\wedge \ \mathsf{inside}(w_3, y_2, w_2, v_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step3c6:

$$
\begin{array}{ll}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_{3a} = w_3x_3) \\
\wedge \ \mathsf{deleting}(y_2, w_2, x_3, w_3) & (d_{3b} = x_3w_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, w_3, w_2) & (d_3(Y) = d_{3a}, e_3 = y_2z_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, x_3, w_2) & (d_3(Y) = d_{3b}, e_3 = y_2x_3) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(w_2, z_1, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, u_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(z_1, w_2, y_2, v_1) & (\leftarrow Step1/2) \\
\wedge \neg \mathsf{inside}(w_3, y_2, w_2, u_1) & (\leftarrow Step3c3) \\
\wedge \ \mathsf{inside}(x_3, y_2, w_2, u_1) & \\
\wedge \ \mathsf{inside}(y_2, w_3, w_2, v_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step4a1:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y, z) & (e_0 = u_1v_1, e_1 = yz) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y, w_2) & (e_2 = yw_2) \\
\wedge \neg \mathsf{inside}(w_2, z, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(w_2, z, y, v_1) & (\leftarrow Step1/2/3) \\
\wedge \ \mathsf{inside}(w_2, z, y, w_3) & \\
\rightarrow \ \mathsf{intersection}(u_1, v_1, y, w_3) & (e_3 = yw_3(Step5/7)) \\
\vee \ \mathsf{intersection}(u_1, v_1, w_3, w_2) & (e_3 = w_3w_2(Step8/9))
\end{array}
$$

Step4a2:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y, z) & (e_0 = u_1v_1, e_1 = yz) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y, w_2) & (e_2 = yw_2) \\
\wedge \neg \mathsf{inside}(w_2, z, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \ \mathsf{inside}(w_2, z, y, w_3) & \\
\wedge \ \mathsf{inside}(w_3, z, y, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step4a3:

$$
\begin{aligned}
&\mathsf{intersection}(u_1, v_1, y, z) && (e_0 = u_1v_1, e_1 = yz)\\
\wedge\ &\mathsf{intersection}(u_1, v_1, y, w_2) && (e_2 = yw_2)\\
\wedge&\neg\mathsf{inside}(w_2, z, y, u_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(w_2, z, y, v_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(z, w_2, y, u_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(z, w_2, y, v_1) && (\leftarrow Step1/2/3)\\
\wedge\ &\mathsf{inside}(w_2, z, y, w_3)\\
\wedge\ &\mathsf{inside}(z, w_3, y, u_1)\\
\rightarrow\ &\bot
\end{aligned}
$$

Step4a4:

$$
\begin{aligned}
&\mathsf{intersection}(u_1, v_1, y, z) && (e_0 = u_1v_1, e_1 = yz)\\
\wedge\ &\mathsf{intersection}(u_1, v_1, y, w_2) && (e_2 = yw_2)\\
\wedge&\neg\mathsf{inside}(w_2, z, y, u_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(w_2, z, y, v_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(z, w_2, y, u_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(z, w_2, y, v_1) && (\leftarrow Step1/2/3)\\
\wedge\ &\mathsf{inside}(w_2, z, y, w_3)\\
\wedge\ &\mathsf{inside}(y, w_3, w_2, u_1)\\
\rightarrow\ &\bot
\end{aligned}
$$

Step4a5:

$$
\begin{aligned}
&\mathsf{intersection}(u_1, v_1, y, z) && (e_0 = u_1v_1, e_1 = yz)\\
\wedge\ &\mathsf{intersection}(u_1, v_1, y, w_2) && (e_2 = yw_2)\\
\wedge&\neg\mathsf{inside}(w_2, z, y, u_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(w_2, z, y, v_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(z, w_2, y, u_1) && (\leftarrow Step1/2/3)\\
\wedge&\neg\mathsf{inside}(z, w_2, y, v_1) && (\leftarrow Step1/2/3)\\
\wedge\ &\mathsf{inside}(w_2, z, y, w_3)\\
\wedge\ &\mathsf{inside}(w_3, y, w_2, u_1)\\
\rightarrow\ &\bot
\end{aligned}
$$

Step4b1:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y, z) & (e_0 = u_1v_1, e_1 = yz) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y, w_2) & (e_2 = yw_2) \\
\wedge \neg \mathsf{inside}(z, w_2, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(z, w_2, y, v_1) & (\leftarrow Step1/2/3) \\
\wedge \ \mathsf{inside}(z, w_2, y, w_3) & \\
\rightarrow \ \mathsf{intersection}(u_1, v_1, y, w_3) & (e_3 = yw_3(Step5/7)) \\
\vee \ \mathsf{intersection}(u_1, v_1, w_3, w_2) & (e_3 = w_3w_2(Step8/9))
\end{array}
$$

Step4b2:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y, z) & (e_0 = u_1v_1, e_1 = yz) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y, w_2) & (e_2 = yw_2) \\
\wedge \neg \mathsf{inside}(z, w_2, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \ \mathsf{inside}(z, w_2, y, w_3) & \\
\wedge \ \mathsf{inside}(z, w_3, y, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step4b3:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y, z) & (e_0 = u_1v_1, e_1 = yz) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y, w_2) & (e_2 = yw_2) \\
\wedge \neg \mathsf{inside}(w_2, z, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(w_2, z, y, v_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(z, w_2, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(z, w_2, y, v_1) & (\leftarrow Step1/2/3) \\
\wedge \ \mathsf{inside}(z, w_2, y, w_3) & \\
\wedge \ \mathsf{inside}(w_3, z, y, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step4b4:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y, z) & (e_0 = u_1v_1, e_1 = yz) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y, w_2) & (e_2 = yw_2) \\
\wedge \neg \mathsf{inside}(w_2, z, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(w_2, z, y, v_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(z, w_2, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(z, w_2, y, v_1) & (\leftarrow Step1/2/3) \\
\wedge \ \mathsf{inside}(z, w_2, y, w_3) & \\
\wedge \ \mathsf{inside}(y, w_3, w_2, u_1) & \\
\rightarrow &
\end{array}
$$

Step4b5:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y, z) & (e_0 = u_1v_1, e_1 = yz) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y, w_2) & (e_2 = yw_2) \\
\wedge \neg \mathsf{inside}(w_2, z, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(w_2, z, y, v_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(z, w_2, y, u_1) & (\leftarrow Step1/2/3) \\
\wedge \neg \mathsf{inside}(z, w_2, y, v_1) & (\leftarrow Step1/2/3) \\
\wedge \ \mathsf{inside}(z, w_2, y, w_3) & \\
\wedge \ \mathsf{inside}(w_3, y, w_2, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step5a:

$$
\begin{array}{ll}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge \ \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2x_2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2w_2) \\
\wedge \ \mathsf{inside}(w_2, z_1, y_2, z_2) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_2) & (e_3 = y_2z_2) \\
\wedge \ \mathsf{deleting}(y_2, z_2, w_4, x_4) & (d_4 = w_4x_4) \\
\wedge \neg \mathsf{deleting}(y_2, z_1, x_4, w_4) & (\leftarrow orderF) \\
\rightarrow \ \mathsf{intersection}(u_1, v_1, y_2, w_4) & (d_3(Z), e_4 = y_2w_4(\rightarrow Step2/4/6)) \\
\vee \ \mathsf{intersection}(u_1, v_1, w_4, z_2) & (d_3(Y), e_4 = w_4z_2(Step4)) \\
\vee \ \mathsf{deleting}(y_2, z_2, x_4, w_4) & (d_{4b} = x_3w_3(Step3))
\end{array}
$$

Step5b:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	deleting(y_2, z_1, w_2, x_2)	$(d_2(Z) = w_2x_2)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	inside(z_1, w_2, y_2, z_2)	
$\wedge$	intersection(u_1, v_1, y_2, z_2)	$(e_3 = y_2z_2)$
$\wedge$	deleting(y_2, z_2, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge\neg$	deleting(y_2, z_1, x_4, w_4)	$(\leftarrow orderF)$
$\rightarrow$	intersection(u_1, v_1, y_2, w_4)	$(d_3(Z), e_4 = y_2w_4(\rightarrow Step2/4/6))$
$\vee$	intersection(u_1, v_1, w_4, z_2)	$(d_3(Y), e_4 = w_4z_2(Step4))$
$\vee$	deleting(y_2, z_2, x_4, w_4)	$(d_{4b} = x_3w_3(Step3))$

Step6:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	deleting(y_2, z_1, w_2, x_2)	$(d_2(Z) = w_2x_2)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	deleting(y_2, w_2, w_3, x_3)	$(d_3(Z) = w_3x_3)$
$\wedge$	intersection(u_1, v_1, y_2, w_3)	$(e_3 = y_2w_3)$
$\wedge$	deleting(y_2, w_3, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge\neg$	deleting(y_2, z_1, x_3, w_3)	$(\leftarrow orderF)$
$\wedge\neg$	deleting(y_2, z_1, x_4, w_4)	$(\leftarrow orderF)$
$\rightarrow$	intersection(u_1, v_1, y_2, w_4)	$(d_4(Z), e_4 = y_2w_4(\rightarrow Step2/4/6))$
$\vee$	intersection(u_1, v_1, w_4, w_3)	$(d_4(Y), e_4 = w_4w_3(\rightarrow Step4))$
$\vee$	deleting(y_2, w_2, x_3, w_3)	$(d_{3b} = x_3w_3(\rightarrow Part2, Step2))$
$\vee$	deleting(y_2, w_3, x_4, w_4)	$(d_{4b} = x_4w_4(\rightarrow Step3))$

Step7a:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	deleting(y_2, z_1, w_2, x_2)	$(d_2(Z) = w_2x_2)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	deleting(y_2, w_2, w_3, x_3)	$(d_3(Z) = w_3x_3)$
$\wedge$	intersection(u_1, v_1, y_2, w_3)	$(e_3 = y_2w_3)$
$\wedge$	inside(y_2, w_2, w_3, z_2)	
$\wedge$	intersection(u_1, v_1, y_2, z_2)	$(e_4 = y_2z_2)$
$\wedge$	deleting(y_2, z_2, w_5, x_5)	$(d_5 = w_5x_5)$
$\wedge\neg$	deleting(y_2, z_1, x_3, w_3)	$(\leftarrow orderF)$
$\wedge\neg$	deleting(y_2, z_1, x_5, w_5)	$(\leftarrow orderF)$
$\rightarrow$	intersection(u_1, v_1, y_2, w_5)	$(d_5(Z), e_5 = y_2w_5(\rightarrow Step2/4/6))$
$\vee$	intersection(u_1, v_1, w_5, w_2)	$(d_5(Y), e_5 = w_5z_2(\rightarrow Step4))$
$\vee$	deleting(y_2, w_2, x_3, w_3)	$(d_{3b} = x_3w_3(\rightarrow Part2, Step2))$
$\vee$	deleting(y_2, z_2, x_5, w_5)	$(d_{5b} = x_5w_5(\rightarrow Step3))$

Step7b:

	deleting(u_1, v_1, y_2, z_1)	$(e_0 = u_1v_1, e_1 = d_1 = y_2z_1)$
$\wedge$	deleting(y_2, z_1, w_2, x_2)	$(d_2(Z) = w_2x_2)$
$\wedge$	intersection(u_1, v_1, y_2, w_2)	$(e_2 = y_2w_2)$
$\wedge$	deleting(y_2, w_2, w_3, x_3)	$(d_3(Z) = w_3x_3)$
$\wedge$	intersection(u_1, v_1, y_2, w_3)	$(e_3 = y_2w_3)$
$\wedge$	inside(y_2, w_3, w_2, z_2)	
$\wedge$	intersection(u_1, v_1, y_2, z_2)	$(e_4 = y_2z_2)$
$\wedge$	deleting(y_2, z_2, w_5, x_5)	$(d_5 = w_5x_5)$
$\wedge\neg$	deleting(y_2, z_1, x_3, w_3)	$(\leftarrow orderF)$
$\wedge\neg$	deleting(y_2, z_1, x_5, w_5)	$(\leftarrow orderF)$
$\rightarrow$	intersection(u_1, v_1, y_2, w_5)	$(d_5(Z), e_5 = y_2w_5(\rightarrow Step2/4/6))$
$\vee$	intersection(u_1, v_1, w_5, w_2)	$(d_5(Y), e_5 = w_5z_2(\rightarrow Step4))$
$\vee$	deleting(y_2, w_2, x_3, w_3)	$(d_{3b} = x_3w_3(\rightarrow Part2, Step2))$
$\vee$	deleting(y_2, z_2, x_5, w_5)	$(d_{5b} = x_5w_5(\rightarrow Step3))$

Step8a1:

$$
\begin{array}{lll}
 & \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge\neg & \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge\neg & \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge & \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge & \mathsf{inside}(z_3, z_1, y_2, w_4) & \\
\rightarrow & \mathsf{intersection}(u_1, v_1, w_4, z_3) & (e_4 = w_4z_3) \\
\vee & \mathsf{intersection}(u_1, v_1, y_3, w_4) & (e_4 = y_3w_4)
\end{array}
$$

Step8a2:

$$
\begin{array}{lll}
 & \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge\neg & \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge\neg & \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge & \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge & \mathsf{inside}(z_3, z_1, y_2, w_4) & \\
\wedge & \mathsf{inside}(w_4, y_2, z_3, u_1) & \\
\rightarrow & \bot &
\end{array}
$$

Step8a3:

$$
\begin{array}{lll}
 & \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge\neg & \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge\neg & \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge & \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge & \mathsf{inside}(z_3, z_1, y_2, w_4) & \\
\wedge & \mathsf{inside}(y_2, w_4, z_3, u_1) & \\
\rightarrow & \bot &
\end{array}
$$

Step8a4:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{inside}(z_3, z_1, y_2, w_4) & \\
\wedge \ \mathsf{inside}(w_4, z_3, y_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step8a5:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{inside}(z_3, z_1, y_2, w_4) & \\
\wedge \ \mathsf{inside}(z_3, w_4, y_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step8b1:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, w_4) & \\
\rightarrow \ \mathsf{intersection}(u_1, v_1, w_4, z_3) & (e_4 = w_4z_3) \\
\vee \ \mathsf{intersection}(u_1, v_1, y_3, w_4) & (e_4 = y_3w_4)
\end{array}
$$

Step8b2:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, w_4) & \\
\wedge \ \mathsf{inside}(w_4, y_2, z_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step8b3:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, w_4) & \\
\wedge \ \mathsf{inside}(y_2, w_4, z_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step8b4:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, w_4) & \\
\wedge \ \mathsf{inside}(z_3, w_4, y_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step8b5:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, w_4) & \\
\wedge \ \mathsf{inside}(w_4, z_3, y_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step9a1:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{intersection}(z_1, z_3, y_3, w_4) & \\
\rightarrow \ \mathsf{intersection}(u_1, v_1, y_3, w_4) & (e_4 = y_3w_4)
\end{array}
$$

Step9a2:

$$
\begin{array}{lr}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{intersection}(z_1, z_3, y_3, w_4) & \\
\wedge \ \mathsf{inside}(w_4, z_3, y_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step9a3:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_3, z_1, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_3, z_1, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{intersection}(z_1, z_3, y_3, w_4) & \\
\wedge \ \mathsf{inside}(z_3, w_4, y_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step9b1:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{intersection}(z_1, z_3, y_3, w_4) & \\
\rightarrow \ \mathsf{intersection}(u_1, v_1, y_3, w_4) & (e_4 = y_3w_4)
\end{array}
$$

Step9b2:

$$
\begin{array}{ll}
\quad \mathsf{intersection}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = y_2z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2z_3) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, u_1) & (\leftarrow Step1 - 9) \\
\wedge \neg \mathsf{inside}(z_1, z_3, y_2, v_1) & (\leftarrow Step1 - 9) \\
\wedge \ \mathsf{inside}(z_1, z_3, y_2, y_3) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_3, z_3) & (e_3 = y_3z_3) \\
\wedge \ \mathsf{intersection}(z_1, z_3, y_3, w_4) & \\
\wedge \ \mathsf{inside}(w_4, z_3, y_3, u_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step9b3:

	$\mathsf{intersection}(u_1, v_1, y_2, z_1)$	$(e_0 = u_1v_1, e_1 = y_2z_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3)$	$(e_2 = y_2z_3)$
$\wedge\neg$	$\mathsf{inside}(z_1, z_3, y_2, u_1)$	$(\leftarrow Step1 - 9)$
$\wedge\neg$	$\mathsf{inside}(z_1, z_3, y_2, v_1)$	$(\leftarrow Step1 - 9)$
$\wedge$	$\mathsf{inside}(z_1, z_3, y_2, y_3)$	
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_3, z_3)$	$(e_3 = y_3z_3)$
$\wedge$	$\mathsf{intersection}(z_1, z_3, y_3, w_4)$	
$\wedge$	$\mathsf{inside}(z_3, w_4, y_3, u_1)$	
$\rightarrow$	$\bot$	

9.4.4 Proof tasks of Part2

Step1

	$\mathsf{intersection}(u_1, v_1, w_1, x_1)$	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	$\mathsf{deleting}(w_1, x_1, w_2, x_2)$	$(d_2 = w_2x_2)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_2, x_1)$	$(e_2 = w_2x_1)$
$\wedge$	$\mathsf{deleting}(w_2, x_1, w_3, x_3)$	$(d_3 = w_3x_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, x_1)$	$(e_3 = w_3x_1)$
$\rightarrow$	$\mathsf{deleting}(w_1, x_1, w_3, x_2)$	$(d'_2 = w_3x_2)$
$\vee$	$\mathsf{deleting}(w_1, x_1, w_3, x_3)$	$(d'_2 = w_3x_3)$
$\vee$	$\mathsf{intersection}(w_3, x_1, w_2, x_2)$	$(\rightarrow Step2)$
$\vee$	$\mathsf{deleting}(w_2, x_1, x_3, w_3)$	$(d_{3b} = x_3w_3(\rightarrow Step3))$
$\vee$	$\mathsf{inside}(w_1, x_1, w_2, w_3)$	$(\rightarrow Step4)$
$\vee$	$\mathsf{inside}(w_1, w_2, x_1, w_3)$	$(\rightarrow Step4)$
$\vee$	$\mathsf{F}(w_2, x_2)$	$(\rightarrow Step5)$

Step2a:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_3 = w_3x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge$	intersection(w_3, x_1, w_2, x_2)	
$\rightarrow$	deleting(w_1, x_1, w_3, x_2)	$(d'_2 = w_3x_2)$
$\vee$	deleting(w_1, x_1, w_3, x_3)	$(d'_2 = w_3x_3)$
$\vee$	deleting(w_2, x_1, x_3, w_3)	$(d_{3b} = x_3w_3(\rightarrow Step2b))$
$\vee$	deleting(w_3, x_1, w_2, x_2)	$(d_4 = w_2x_2(\rightarrow Step2c))$

Step2b:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_{3a} = w_3x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(d_3 = d_{3a}, e_3 = w_3x_1)$
$\wedge$	deleting(w_2, x_1, x_3, w_3)	$(d_{3b} = x_3w_3)$
$\wedge$	intersection(u_1, v_1, x_3, x_1)	$(d_3 = d_{3b}, e_3 = x_3x_1)$
$\wedge$	intersection(w_3, x_1, w_2, x_2)	$(d_3 = d_{3a}, d_4 = d_2 = w_2x_2)$
$\rightarrow$	deleting(w_1, x_1, w_3, x_2)	$(d'_2 = w_3x_2)$
$\vee$	deleting(w_1, x_1, w_3, x_3)	$(d'_2 = w_3x_3)$
$\vee$	deleting(w_1, x_1, x_3, x_2)	$(d'_2 = x_3x_2)$
$\vee$	deleting(w_1, x_1, x_3, w_3)	$(d'_2 = x_3w_3)$

Step2c:

	deleting(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = d_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_3 = w_3x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge$	deleting(w_3, x_1, w_2, x_2)	$(d_4 = d_2 = w_2x_2)$
$\wedge\neg$	deleting(w_2, x_1, x_3, w_3)	$(\leftarrow Step2b)$
$\rightarrow$	deleting(w_1, x_1, x_2, w_2)	$(d_{2b} = x_2w_2(\rightarrow Step6))$

Step3

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_{3a} = w_3x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(d_3 = d_{3a}, e_3 = w_3x_1)$
$\wedge$	deleting(w_2, x_1, x_3, w_3)	$(d_{3b} = x_3w_3)$
$\wedge$	intersection(u_1, v_1, x_3, x_1)	$(d_3 = d_{3b}, e_3 = x_3x_1)$
$\rightarrow$	deleting(w_1, x_1, w_3, x_2)	$(d'_2 = w_3x_2)$
$\vee$	deleting(w_1, x_1, w_3, x_3)	$(d'_2 = w_3x_3)$
$\vee$	deleting(w_1, x_1, x_3, w_3)	$(d'_2 = x_3w_3)$
$\vee$	deleting(w_1, x_1, x_3, x_2)	$(d'_2 = x_3x_2)$
$\vee$	inside(w_1, x_1, w_2, w_3)	$(\rightarrow Step4)$
$\vee$	inside(w_1, x_1, w_2, x_3)	$(\rightarrow Step4)$
$\vee$	inside(x_1, w_1, w_2, w_3)	$(\rightarrow Step4)$
$\vee$	inside(x_1, w_1, w_2, x_3)	$(\rightarrow Step4)$

Step4a1:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3/4)$
$\wedge$	inside(w_1, x_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(e_4 = w_4x_1)$
$\wedge$	intersection(w_4, x_1, w_2, x_2)	
$\rightarrow$	deleting(w_1, x_1, w_4, x_2)	$(d'_2 = w_4x_2)$
$\vee$	deleting(w_1, x_1, w_4, x_4)	$(d'_2 = w_4x_4)$
$\vee$	deleting(w_2, x_1, w_4, x_2)	$(d'_3 = w_4x_2)$
$\vee$	deleting(w_2, x_1, w_4, x_4)	$(d'_3 = w_4x_4)$
$\vee$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4(Step4a2))$

Step4a2:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3)$
$\wedge$	inside(w_1, x_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_{4a} = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(d_4 = d_{4a}, e_4 = w_4x_1)$
$\wedge$	intersection(w_4, x_1, w_2, x_2)	
$\wedge$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4)$
$\wedge$	intersection(u_1, v_1, x_4, x_1)	$(d_4 = d_{4b}, e_4 = x_4x_1)$
$\wedge$	intersection(x_4, x_1, w_2, x_2)	
$\rightarrow$	$\bot$	

Step4a3:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_3 = w_3x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge$	inside(w_1, x_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(e_4 = w_4x_1)$
$\wedge$	deleting(w_4, x_1, x_3, w_3)	$(d_5 = x_3w_3)$
$\rightarrow$	deleting(w_1, x_1, x_3, x_4)	$(d'_2 = x_3x_4)$
$\vee$	deleting(w_2, x_1, x_3, x_4)	$(d'_3 = x_3x_4)$

Step4b1:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3/4)$
$\wedge$	inside(w_1, x_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(e_4 = w_4x_1)$
$\rightarrow$	deleting(w_1, x_1, w_4, x_2)	$(d'_2 = w_4x_2)$
$\vee$	deleting(w_1, x_1, w_4, x_4)	$(d'_2 = w_4x_4)$
$\vee$	deleting(w_2, x_1, w_4, x_2)	$(d'_3 = w_4x_2)$
$\vee$	deleting(w_2, x_1, w_4, x_4)	$(d'_3 = w_4x_4)$
$\vee$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4(Step4b2))$
$\vee$	inside(w_1, x_1, w_2, w_4)	$(w_4\ as\ w_3\ in\ Step4)$

Step4b2:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3)$
$\wedge$	inside(w_1, x_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_{4a} = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(d_4 = d_{4a}, e_4 = w_4x_1)$
$\wedge$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4)$
$\wedge$	intersection(u_1, v_1, x_4, x_1)	$(d_4 = d_{4b}, e_4 = x_4x_1)$
$\rightarrow$	deleting(w_1, x_1, w_4, x_4)	$(d'_2 = w_4x_4)$
$\vee$	deleting(w_1, x_1, x_4, w_4)	$(d'_2 = x_4w_4)$
$\vee$	deleting(w_2, x_1, w_4, x_4)	$(d'_3 = w_4x_4)$
$\vee$	deleting(w_2, x_1, x_4, w_4)	$(d'_3 = x_4w_4)$
$\vee$	inside(w_1, x_1, w_2, w_4)	$(w_4\ as\ w_3\ in\ Step4b)$
$\vee$	inside(w_1, x_1, w_2, x_4)	$(x_4\ as\ w_3\ in\ Step4b)$

Step4c1:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3)$
$\wedge$	inside(x_1, w_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(e_4 = w_4x_1)$
$\wedge$	intersection(w_4, x_1, w_2, x_2)	
$\rightarrow$	deleting(w_1, x_1, w_4, x_2)	$(d'_2 = w_4x_2)$
$\vee$	deleting(w_1, x_1, w_4, x_4)	$(d'_2 = w_4x_4)$
$\vee$	deleting(w_2, x_1, w_4, x_2)	$(d'_3 = w_4x_2)$
$\vee$	deleting(w_2, x_1, w_4, x_4)	$(d'_3 = w_4x_4)$
$\vee$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4(Step4c2))$

Step4c2:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3)$
$\wedge$	inside(x_1, w_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_{4a} = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(d_4 = d_{4a}, e_4 = w_4x_1)$
$\wedge$	intersection(w_4, x_1, w_2, x_2)	
$\wedge$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4)$
$\wedge$	intersection(u_1, v_1, x_4, x_1)	$(d_4 = d_{4b}, e_4 = x_4x_1)$
$\wedge$	intersection(x_4, x_1, w_2, x_2)	
$\rightarrow$	$\bot$	

Step4c3:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_3 = w_3x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge$	inside(x_1, w_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(e_4 = w_4x_1)$
$\wedge$	deleting(w_4, x_1, x_3, w_3)	$(d_5 = x_3w_3)$
$\rightarrow$	deleting(w_1, x_1, x_3, x_4)	$(d'_2 = x_3x_4)$
$\vee$	deleting(w_2, x_1, x_3, x_4)	$(d'_3 = x_3x_4)$

Step4d1:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3)$
$\wedge$	inside(x_1, w_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(e_4 = w_4x_1)$
$\rightarrow$	deleting(w_1, x_1, w_4, x_2)	$(d'_2 = w_4x_2)$
$\vee$	deleting(w_1, x_1, w_4, x_4)	$(d'_2 = w_4x_4)$
$\vee$	deleting(w_2, x_1, w_4, x_2)	$(d'_3 = w_4x_2)$
$\vee$	deleting(w_2, x_1, w_4, x_4)	$(d'_3 = w_4x_4)$
$\vee$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4(Step4d2))$
$\vee$	inside(x_1, w_1, w_2, w_4)	$(w_4\ as\ w_3\ in\ Step4c)$

Step4d2:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3)$
$\wedge$	inside(x_1, w_1, w_2, w_3)	
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_{4a} = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(d_4 = d_{4a}, e_4 = w_4x_1)$
$\wedge$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4)$
$\wedge$	intersection(u_1, v_1, x_4, x_1)	$(d_4 = d_{4b}, e_4 = x_4x_1)$
$\rightarrow$	deleting(w_1, x_1, w_4, x_4)	$(d'_2 = w_4x_4)$
$\vee$	deleting(w_1, x_1, x_4, w_4)	$(d'_2 = x_4w_4)$
$\vee$	deleting(w_2, x_1, w_4, x_4)	$(d'_3 = w_4x_4)$
$\vee$	deleting(w_2, x_1, x_4, w_4)	$(d'_3 = x_4w_4)$
$\vee$	inside(x_1, w_1, w_2, w_4)	$(w_4\ as\ w_3\ in\ Step4d)$
$\vee$	inside(x_1, w_1, w_2, x_4)	$(x_4\ as\ w_3\ in\ Step4d)$

Step5:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(e_1 = w_1x_1, d_2 = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_3 = w_3x_3)$
$\wedge\neg$	deleting(w_1, x_1, w_3, x_2)	$(shortest\ cycle)$
$\wedge\neg$	deleting(w_1, x_1, w_3, x_3)	$(shortest\ cycle)$
$\wedge\neg$	deleting(w_1, x_1, x_3, w_3)	$(\leftarrow orderF)$
$\wedge\neg$	deleting(w_2, x_1, x_3, w_3)	$(\leftarrow Step2)$
$\wedge\neg$	intersection(w_3, x_1, w_2, x_2)	$(\leftarrow Step3)$
$\wedge\neg$	inside(w_1, x_1, w_2, w_3)	$(\leftarrow Step4)$
$\wedge\neg$	inside(x_1, w_1, w_2, w_3)	$(\leftarrow Step4)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(e_3 = w_3x_1)$
$\wedge$	deleting(w_3, x_1, w_4, x_4)	$(d_4 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(e_4 = w_4x_1)$
$\rightarrow$	deleting(w_2, x_1, w_4, x_3)	$(d'_3 = w_4x_3)$
$\vee$	deleting(w_2, x_1, w_4, x_4)	$(d'_3 = w_4x_4)$
$\vee$	deleting(w_3, x_1, x_4, w_4)	$(d_{4b} = x_4w_4(\rightarrow Step2))$
$\vee$	intersection(w_4, x_1, w_3, x_3)	$(\rightarrow Step3)$
$\vee$	inside(x_1, w_2, w_3, w_4)	$(\rightarrow Step4)$
$\vee$	inside(x_1, w_3, w_2, w_4)	$(\rightarrow Step4)$

Step6a1:

intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$ deleting(w_1, x_1, w_2, x_2)	$(d_{2a} = w_2x_2)$
$\wedge$ intersection(u_1, v_1, w_2, x_1)	$(d_2 = d_{2a}, e_2 = w_2x_1)$
$\wedge$ deleting(w_2, x_1, w_3, x_3)	$(d_2 = d_{2a}, d_3 = w_1x_3)$
$\wedge$ intersection(u_1, v_1, w_3, x_1)	$(d_2 = d_{2a}, e_3 = w_3x_1)$
$\wedge$ deleting(w_3, x_1, w_2, x_2)	$(d_2 = d_{2a}, d_4 = w_2x_2)$
$\wedge$ deleting(w_1, x_1, x_2, w_2)	$(d_{2b} = x_2w_2)$
$\wedge$ intersection(u_1, v_1, x_2, x_1)	$(d_2 = d_{2b}, e_2 = x_2x_1)$
$\wedge$ deleting(x_2, x_1, w_4, x_4)	$(d_2 = d_{2b}, d_3 = w_4x_4)$
$\wedge$ intersection(u_1, v_1, w_4, x_1)	$(d_2 = d_{2b}, e_3 = w_4x_1)$
$\wedge$ intersection(w_4, x_1, w_2, x_2)	
$\rightarrow \bot$	

Step6a2:

intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$ deleting(w_1, x_1, w_2, x_2)	$(d_{2a} = w_2x_2)$
$\wedge$ intersection(u_1, v_1, w_2, x_1)	$(d_2 = d_{2a}, e_2 = w_2x_1)$
$\wedge$ deleting(w_2, x_1, w_3, x_3)	$(d_2 = d_{2a}, d_3 = w_1x_3)$
$\wedge$ intersection(u_1, v_1, w_3, x_1)	$(d_2 = d_{2a}, e_3 = w_3x_1)$
$\wedge$ deleting(w_3, x_1, w_2, x_2)	$(d_2 = d_{2a}, d_4 = w_2x_2)$
$\wedge$ deleting(w_1, x_1, x_2, w_2)	$(d_{2b} = x_2w_2)$
$\wedge$ intersection(u_1, v_1, x_2, x_1)	$(d_2 = d_{2b}, e_2 = x_2x_1)$
$\wedge$ deleting(x_2, x_1, w_2, x_4)	$(d_2 = d_{2b}, d_3 = w_2x_4)$
$\wedge$ intersection(u_1, v_1, w_2, x_1)	$(d_2 = d_{2b}, e_3 = w_2x_1)$
$\rightarrow$ F(w_2, x_4)	$(contradiction,$ w_2x_2 *is not closest* $F)$

Step6b1:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_{2a} = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(d_2 = d_{2a}, e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_2 = d_{2a}, d_3 = w_1x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(d_2 = d_{2a}, e_3 = w_3x_1)$
$\wedge$	deleting(w_3, x_1, w_2, x_2)	$(d_2 = d_{2a}, d_4 = w_2x_2)$
$\wedge$	deleting(w_1, x_1, x_2, w_2)	$(d_{2b} = x_2w_2)$
$\wedge$	intersection(u_1, v_1, x_2, x_1)	$(d_2 = d_{2b}, e_2 = x_2x_1)$
$\wedge$	deleting(x_2, x_1, w_4, x_4)	$(d_2 = d_{2b}, d_3 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(d_2 = d_{2b}, e_3 = w_4x_1)$
$\wedge$	inside(w_1, x_1, x_2, w_4)	
$\wedge$	deleting(w_4, x_1, w_5, x_5)	$(d_2 = d_{2b}, d_4 = w_5x_5)$
$\wedge$	intersection(u_1, v_1, w_5, x_1)	$(d_2 = d_{2b}, e_4 = w_5x_1)$
$\wedge\neg$	deleting(w_5, x_1, x_4, w_4)	$(\leftarrow Step4a3)$
$\wedge$	intersection(w_5, x_1, w_2, x_2)	
$\rightarrow$	$\perp$	

Step6b2:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_{2a} = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(d_2 = d_{2a}, e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_2 = d_{2a}, d_3 = w_1x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(d_2 = d_{2a}, e_3 = w_3x_1)$
$\wedge$	deleting(w_3, x_1, w_2, x_2)	$(d_2 = d_{2a}, d_4 = w_2x_2)$
$\wedge$	deleting(w_1, x_1, x_2, w_2)	$(d_{2b} = x_2w_2)$
$\wedge$	intersection(u_1, v_1, x_2, x_1)	$(d_2 = d_{2b}, e_2 = x_2x_1)$
$\wedge$	deleting(x_2, x_1, w_4, x_4)	$(d_2 = d_{2b}, d_3 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(d_2 = d_{2b}, e_3 = w_4x_1)$
$\wedge$	inside(w_1, x_1, x_2, w_4)	
$\wedge$	deleting(w_4, x_1, w_2, x_5)	$(d_2 = d_{2b}, d_4 = w_2x_5)$
$\rightarrow$	F(w_2, x_5)	$(contradiction,$ w_2x_2 *is not closest* $F)$

Step6c1:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_{2a} = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(d_2 = d_{2a}, e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_2 = d_{2a}, d_3 = w_1x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(d_2 = d_{2a}, e_3 = w_3x_1)$
$\wedge$	deleting(w_3, x_1, w_2, x_2)	$(d_2 = d_{2a}, d_4 = w_2x_2)$
$\wedge$	deleting(w_1, x_1, x_2, w_2)	$(d_{2b} = x_2w_2)$
$\wedge$	intersection(u_1, v_1, x_2, x_1)	$(d_2 = d_{2b}, e_2 = x_2x_1)$
$\wedge$	deleting(x_2, x_1, w_4, x_4)	$(d_2 = d_{2b}, d_3 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(d_2 = d_{2b}, e_3 = w_4x_1)$
$\wedge$	inside(x_1, w_1, x_2, w_4)	
$\wedge$	deleting(w_4, x_1, w_5, x_5)	$(d_2 = d_{2b}, d_4 = w_5x_5)$
$\wedge$	intersection(u_1, v_1, w_5, x_1)	$(d_2 = d_{2b}, e_4 = w_5x_1)$
$\wedge\neg$	deleting(w_5, x_1, x_4, w_4)	$(\leftarrow Step4c3)$
$\wedge$	intersection(w_5, x_1, w_2, x_2)	
$\rightarrow$	$\bot$	

Step6c2:

	intersection(u_1, v_1, w_1, x_1)	$(e_0 = u_1v_1, e_1 = w_1x_1)$
$\wedge$	deleting(w_1, x_1, w_2, x_2)	$(d_{2a} = w_2x_2)$
$\wedge$	intersection(u_1, v_1, w_2, x_1)	$(d_2 = d_{2a}, e_2 = w_2x_1)$
$\wedge$	deleting(w_2, x_1, w_3, x_3)	$(d_2 = d_{2a}, d_3 = w_1x_3)$
$\wedge$	intersection(u_1, v_1, w_3, x_1)	$(d_2 = d_{2a}, e_3 = w_3x_1)$
$\wedge$	deleting(w_3, x_1, w_2, x_2)	$(d_2 = d_{2a}, d_4 = w_2x_2)$
$\wedge$	deleting(w_1, x_1, x_2, w_2)	$(d_{2b} = x_2w_2)$
$\wedge$	intersection(u_1, v_1, x_2, x_1)	$(d_2 = d_{2b}, e_2 = x_2x_1)$
$\wedge$	deleting(x_2, x_1, w_4, x_4)	$(d_2 = d_{2b}, d_3 = w_4x_4)$
$\wedge$	intersection(u_1, v_1, w_4, x_1)	$(d_2 = d_{2b}, e_3 = w_4x_1)$
$\wedge$	inside(x_1, w_1, x_2, w_4)	
$\wedge$	deleting(w_4, x_1, w_2, x_5)	$(d_2 = d_{2b}, d_4 = w_2x_5)$
$\rightarrow$	F(w_2, x_5)	$(contradiction,$ w_2x_2 *is not closest* $F)$

9.4.5 Proof tasks of Part3

Step1a1:

$$
\begin{array}{ll}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\rightarrow & \mathsf{F}(w_3, x_3) & (Step1b) \\
\vee & \mathsf{inside}(z_1, z_3, y_2, w_3) \vee \mathsf{inside}(z_3, z_1, y_2, w_3) & (Step2) \\
\vee & \mathsf{intersection}(u_1, v_1, w_3, x_3) & (Step3)
\end{array}
$$

Step1a2:

$$
\begin{array}{ll}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\wedge & \mathsf{intersection}(y_2, z_1, w_3, z_3) & \\
\rightarrow & \bot &
\end{array}
$$

Step1b1:

$$
\begin{array}{ll}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3x_3) \\
\wedge & \mathsf{F}(w_3, x_3) & \\
\rightarrow & \mathsf{deleting}(y_2, z_3, x_3, w_3) & (d_{3b}(Y) = x_3w_3(Step1b2)) \\
\vee & \mathsf{inside}(z_1, z_3, y_2, w_3) \vee \mathsf{inside}(z_3, z_1, y_2, w_3) & (Step2)
\end{array}
$$

Step1b2:

$$
\begin{array}{ll}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1v_1, e_1 = d_1 = y_2z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_{3a}(Y) = w_3x_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, x_3, w_3) & (d_{3b}(Y) = x_3w_3) \\
\rightarrow & \mathsf{inside}(z_1, z_3, y_2, w_3) \vee \mathsf{inside}(z_3, z_1, y_2, w_3) & (Step2) \\
\vee & \mathsf{inside}(z_1, z_3, y_2, x_3) \vee \mathsf{inside}(z_3, z_1, y_2, x_3) & (Step2)
\end{array}
$$

Step1c1:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, z_3, x_2) && (d_2(Z) = z_3 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_3, x_3) && (d_3(Y) = w_3 x_3) \\
\wedge\ & \mathsf{inside}(w_3, y_2, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1c2:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_2, x_2) && (d_2(Z) = w_2 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_2) && (e_2 = y_2 w_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(w_2, z_1, y_2, z_3) && (e_3 = y_2 z_3) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_4, x_4) && (d_4(Y) = w_4 x_4) \\
\wedge\ & \mathsf{inside}(w_4, y_2, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1c3:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_2, x_2) && (d_2(Z) = w_2 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_2) && (e_2 = y_2 w_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(z_1, w_2, y_2, z_3) && (e_3 = y_2 z_3) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_4, x_4) && (d_4(Y) = w_4 x_4) \\
\wedge\ & \mathsf{inside}(w_4, y_2, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1c4:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_2, x_2) && (d_2(Z) = w_2 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_2) && (e_2 = y_2 w_2) \\
\wedge\ & \mathsf{deleting}(y_2, w_2, z_3, x_3) && (d_3(Z) = z_3 x_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) && (e_3 = y_2 z_3) \\
\wedge & \neg\mathsf{deleting}(y_2, w_2, x_3, z_3) && (\leftarrow Part2, Step2) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_4, x_4) && (d_4(Y) = w_4 x_4) \\
\wedge\ & \mathsf{inside}(w_4, y_2, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1c5:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_2, x_2) && (d_2(Z) = w_2 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_2) && (e_2 = y_2 w_2) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_3, x_3) && (d_3(Z) = w_3 x_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_3) && (e_3 = y_2 w_3) \\
\wedge & \neg\mathsf{deleting}(y_2, w_2, x_3, w_3) && (\leftarrow Part2, Step2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(y_2, w_2, w_3, z_3) && (e_4 = y_2 z_3) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_5, x_5) && (d_5(Y) = w_5 x_5) \\
\wedge\ & \mathsf{inside}(w_5, y_2, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1c6:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_2, x_2) && (d_2(Z) = w_2 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_2) && (e_2 = y_2 w_2) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_3, x_3) && (d_3(Z) = w_3 x_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_3) && (e_3 = y_2 w_3) \\
\wedge & \neg\mathsf{deleting}(y_2, w_2, x_3, w_3) && (\leftarrow Part2, Step2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(y_2, w_3, w_2, z_3) && (e_4 = y_2 z_3) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_5, x_5) && (d_5(Y) = w_5 x_5) \\
\wedge\ & \mathsf{inside}(w_5, y_2, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1d1:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, z_3, x_2) && (d_2(Z) = z_3 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_3, x_3) && (d_3(Y) = w_3 x_3) \\
\wedge\ & \mathsf{inside}(y_2, w_3, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1d2:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_2, x_2) && (d_2(Z) = w_2 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_2) && (e_2 = y_2 w_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(w_2, z_1, y_2, z_3) && (e_3 = y_2 z_3) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_4, x_4) && (d_4(Y) = w_4 x_4) \\
\wedge\ & \mathsf{inside}(y_2, w_4, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1d3:

$$
\begin{aligned}
& \mathsf{deleting}(u_1, v_1, y_2, z_1) && (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge\ & \mathsf{deleting}(y_2, z_1, w_2, x_2) && (d_2(Z) = w_2 x_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, w_2) && (e_2 = y_2 w_2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(z_1, w_2, y_2, z_3) && (e_3 = y_2 z_3) \\
\wedge\ & \mathsf{deleting}(y_2, z_3, w_4, x_4) && (d_4(Y) = w_4 x_4) \\
\wedge\ & \mathsf{inside}(y_2, w_4, z_3, z_1) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step1d4:

$$
\begin{array}{lr}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge \ \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2 x_2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2 w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, z_3, x_3) & (d_3(Z) = z_3 x_3) \\
\wedge \neg \mathsf{deleting}(y_2, w_2, x_3, z_3) & (\leftarrow Part2, Step2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_3 = y_2 z_3) \\
\wedge \ \mathsf{deleting}(y_2, z_3, w_4, x_4) & (d_4(Y) = w_4 x_4) \\
\wedge \ \mathsf{inside}(y_2, w_4, z_3, z_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step1d5:

$$
\begin{array}{lr}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge \ \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2 x_2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2 w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3(Z) = w_3 x_3) \\
\wedge \neg \mathsf{deleting}(y_2, w_2, x_3, w_3) & (\leftarrow Part2, Step2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2 w_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(y_2, w_2, w_3, z_3) & (e_4 = y_2 z_3) \\
\wedge \ \mathsf{deleting}(y_2, z_3, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge \ \mathsf{inside}(y_2, w_5, z_3, z_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step1d6:

$$
\begin{array}{lr}
\quad \mathsf{deleting}(u_1, v_1, y_2, z_1) & (e_0 = u_1 v_1, e_1 = d_1 = y_2 z_1) \\
\wedge \ \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2 x_2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2 w_2) \\
\wedge \ \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3(Z) = w_3 x_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2 w_3) \\
\wedge \neg \mathsf{deleting}(y_2, w_2, x_3, w_3) & (\leftarrow Part2, Step2) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(y_2, w_3, w_2, z_3) & (e_4 = y_2 z_3) \\
\wedge \ \mathsf{deleting}(y_2, z_3, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge \ \mathsf{inside}(y_2, w_5, z_3, z_1) & \\
\rightarrow \ \bot &
\end{array}
$$

Step2a1:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\rightarrow & \mathsf{F}(w_5, x_5) & (Step2b1) \\
\vee & \mathsf{inside}(z_5, z_3, y_4, w_5) & (Step2c) \\
\vee & \mathsf{inside}(y_4, y_2, z_5, w_5) & (Step3) \\
\vee & \mathsf{intersection}(y_2, y_4, w_5, z_5) & (Step3)
\end{array}
$$

Step2a2:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) \wedge \neg \mathsf{F}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge & \mathsf{intersection}(y_4, z_3, w_5, z_5) & \\
\rightarrow & \bot &
\end{array}
$$

Step2b1:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge & \mathsf{F}(w_5, x_5) & \\
\rightarrow & \mathsf{deleting}(y_4, z_5, x_5, w_5) & (d_{5b}(Y) = x_5 w_5 (Step2b2)) \\
\vee & \mathsf{inside}(z_5, z_3, y_4, w_5) & (Step2c)
\end{array}
$$

Step2b2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5)$	$(e_4 = y_4 z_5)$
$\wedge$	$\mathsf{deleting}(y_4, z_5, w_5, x_5)$	$(d_{5a}(Y) = w_5 x_5)$
$\wedge$	$\mathsf{deleting}(y_4, z_5, x_5, w_5)$	$(d_{5b}(Y) = x_5 w_5)$
$\rightarrow$	$\mathsf{inside}(z_5, z_3, y_4, w_5)$	$(Step2c)$
$\vee$	$\mathsf{inside}(z_5, z_3, y_4, x_5)$	$(Step2c)$

Step2c1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1)$	$(e_1 = y_2 z_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3)$	$(e_3 = y_3 z_3)$
$\wedge$	$\mathsf{deleting}(y_3, z_3, w_4, x_4)$	$(d_4 = w_4 x_4)$
$\rightarrow$	$\mathsf{deleting}(y_2, z_3, w_4, x_4)$	$(d'_3(Y) = w_4 x_4 (Step1/Step2))$
$\vee$	$\mathsf{deleting}(y_3, z_3, x_4, w_4)$	$(d_{4b} = x_4 w_4 (Step2d))$
$\vee$	$\mathsf{deleting}(y_3, z_3, y_2, x_4)$	$(d'_4(Y) = y_2 x_4 (Step2e))$
$\vee$	$\mathsf{inside}(z_3, z_1, y_2, w_4)$	$(d_4(Y) : Step2c\ with\ w_4\ as\ y_3,$ $d_4(Z) : Step6)$
$\vee$	$\mathsf{intersection}(u_1, v_1, y_3, w_4)$	$(d_4(Z) : Step6)$

Step2c2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1)$	$(e_1 = y_2 z_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3)$	$(e_3 = y_3 z_3)$
$\wedge\neg$	$\mathsf{intersection}(y_2, z_1, y_3, z_3)$	$(\leftarrow Step1a2/Step2a3)$
$\wedge$	$\mathsf{deleting}(y_3, z_3, w_4, x_4)$	$(d_4(Y) = w_4 x_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_4, z_3)$	$(e_4 = w_4 z_3)$
$\wedge$	$\mathsf{intersection}(y_2, z_1, w_4, z_3)$	
$\rightarrow$	$\mathsf{deleting}(y_3, z_3, y_2, x_4)$	$(d'_4(Y) = y_2 x_4 (Step2e))$

Step2d1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) \qquad\qquad (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) \qquad\qquad (e_3 = y_3 z_3) \\
\wedge & \mathsf{deleting}(y_3, z_3, w_4, x_4) \qquad\qquad (d_{4a} = w_4 x_4) \\
\wedge & \mathsf{deleting}(y_3, z_3, x_4, w_4) \qquad\qquad (d_{4b} = x_4 w_4) \\
\rightarrow & \mathsf{inside}(z_3, z_1, y_2, w_4) \qquad\qquad (d_4(Y) : Step2c \ with \ w_4 \ as \ y_3, \\
 & \qquad\qquad d_4(Z) : Step6) \\
\vee & \mathsf{inside}(z_3, z_1, y_2, x_4) \qquad\qquad (d_4(Y) : Step2c \ with \ x_4 \ as \ y_3, \\
 & \qquad\qquad d_4(Z) : Step6)
\end{array}
$$

Step2d2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) \qquad\qquad (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) \qquad\qquad (d_{3a}(Y) = w_3 x_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, x_3, w_3) \qquad\qquad (d_{3b}(Y) = x_3 w_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, w_3) \qquad\qquad (d_3 = d_{3a}, e_3 = w_3 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, x_3, z_3) \qquad\qquad (d_3 = d_{3b}, e_3 = x_3 z_3) \\
\wedge & \mathsf{deleting}(w_3, z_3, w_4, x_4) \qquad\qquad (d_3 = d_{3a}, d_4(Y) = w_4 x_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_4, z_3) \qquad\qquad (d_3 = d_{3a}, e_4 = w_4 z_3) \\
\wedge & \mathsf{deleting}(w_4, z_3, w_3, x_3) \qquad\qquad (d_3 = d_{3a}, d_5(Y) = w_3 x_3) \\
\rightarrow & \mathsf{deleting}(y_2, z_3, x_3, x_4) \qquad\qquad (d_3'(Y) = x_3 x_4 (Step1/Step2)) \\
\vee & \mathsf{deleting}(w_3, z_3, y_2, x_4) \qquad\qquad (d_3 = d_{3a}, d_4'(Y) = y_2 x_4 (Step2e))
\end{array}
$$

Step2d3:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) & (e_3 = y_3 z_3) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_3, z_3) & (Step1a2/Step2a3) \\
\wedge & \mathsf{deleting}(y_3, z_3, w_4, x_4) & (d_{4a}(Y) = w_4 x_4) \\
\wedge & \mathsf{deleting}(y_3, z_3, x_4, w_4) & (d_{4b}(Y) = x_4 w_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_4, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, w_4) & (d_4 = d_{4a}, e_4 = w_4 z_3) \\
\wedge & \mathsf{intersection}(y_2, z_1, w_4, z_3) & \\
\rightarrow & \bot &
\end{array}
$$

Step2e1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, z_3, x_2) & (d_2(Z) = z_3 x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) & (e_3 = y_3 z_3) \\
\wedge & \mathsf{deleting}(y_3, z_3, y_2, x_4) & (d_4(Y) = y_2 x_4) \\
\wedge & \neg\mathsf{deleting}(y_3, z_3, x_4, y_2) & (\leftarrow (Step2d1)) \\
\rightarrow & \bot &
\end{array}
$$

Step2e2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2 x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2 w_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(w_2, z_1, y_2, z_3) & (e_3 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) & (e_4 = y_3 z_3) \\
\wedge & \mathsf{deleting}(y_3, z_3, y_2, x_5) & (d_5(Y) = y_2 x_5) \\
\wedge & \neg\mathsf{deleting}(y_3, z_3, x_5, y_2) & (\leftarrow (Step2d1)) \\
\rightarrow & \bot &
\end{array}
$$

Step2e3:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2 x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2 w_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(z_1, w_2, y_2, z_3) & (e_3 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) & (e_4 = y_3 z_3) \\
\wedge & \mathsf{deleting}(y_3, z_3, y_2, x_5) & (d_5(Y) = y_2 x_5) \\
\wedge & \neg\mathsf{deleting}(y_3, z_3, x_5, y_2) & (\leftarrow (Step2d1)) \\
\rightarrow & \bot &
\end{array}
$$

Step2e4:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2 x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2 w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, z_3, x_3) & (d_3(Z) = z_3 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) & (e_3 = y_2 z_3) \\
\wedge & \neg\mathsf{deleting}(y_2, w_2, x_3, z_3) & (\leftarrow Part2, Step2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) & (e_4 = y_3 z_3) \\
\wedge & \mathsf{deleting}(y_3, z_3, y_2, x_5) & (d_5(Y) = y_2 x_5) \\
\wedge & \neg\mathsf{deleting}(y_3, z_3, x_5, y_2) & (\leftarrow (Step2d1)) \\
\rightarrow & \bot &
\end{array}
$$

Step2e5:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2 x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2 w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3(Z) = w_3 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2 w_3) \\
\wedge & \neg\mathsf{deleting}(y_2, w_2, x_3, w_3) & (\leftarrow Part2, Step2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(y_2, w_2, w_3, z_3) & (e_4 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) & (e_5 = y_3 z_3) \\
\wedge & \mathsf{deleting}(y_3, z_3, y_2, x_6) & (d_6(Y) = y_2 x_6) \\
\wedge & \neg\mathsf{deleting}(y_3, z_3, x_6, y_2) & (\leftarrow (Step2d1)) \\
\rightarrow & \bot &
\end{array}
$$

Step2e6:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{deleting}(y_2, z_1, w_2, x_2) & (d_2(Z) = w_2 x_2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_2) & (e_2 = y_2 w_2) \\
\wedge & \mathsf{deleting}(y_2, w_2, w_3, x_3) & (d_3(Z) = w_3 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, w_3) & (e_3 = y_2 w_3) \\
\wedge & \neg\mathsf{deleting}(y_2, w_2, x_3, w_3) & (\leftarrow Part2, Step2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{inside}(y_2, w_3, w_2, z_3) & (e_4 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_3) & (e_5 = y_3 z_3) \\
\wedge & \mathsf{deleting}(y_3, z_3, y_2, x_6) & (d_6(Y) = y_2 x_6) \\
\wedge & \neg\mathsf{deleting}(y_3, z_3, x_6, y_2) & (\leftarrow (Step2d1)) \\
\rightarrow & \bot &
\end{array}
$$

Step3a1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, w_3, x_3)$	$(d_3(Y) = w_3 x_3)$
$\wedge\neg$	$\mathsf{F}(w_3, x_3)$	$(\leftarrow Step1b/2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, z_3)$	$(e_3 = w_3 z_3)$
$\wedge$	$\mathsf{deleting}(w_3, z_3, w_4, x_4)$	$(d_4 = w_4 x_4)$
$\rightarrow$	$\mathsf{deleting}(y_2, z_3, w_4, x_3)$	$(d_3'(Y) = w_4 x_3(Step1/Step2))$
$\vee$	$\mathsf{deleting}(y_2, z_3, w_4, x_4)$	$(d_3'(Y) = w_4 x_4(Step1/Step2))$
$\vee$	$\mathsf{intersection}(w_3, x_3, w_4, z_3)$	$(Step3a2)$
$\vee$	$\mathsf{deleting}(w_3, z_3, x_4, w_4)$	$(d_{4b} = x_4 w_4(Step3a3))$
$\vee$	$\mathsf{inside}(y_2, w_3, z_3, w_4) \vee \mathsf{inside}(w_3, y_2, z_3, w_4)$	$(d_4(Y) : Step3b,$
		$d_4(Z) : Step4a)$

Step3a2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, w_3, x_3)$	$(d_3(Y) = w_3 x_3)$
$\wedge\neg$	$\mathsf{F}(w_3, x_3)$	$(\leftarrow Step1b/2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, z_3)$	$(e_3 = w_3 z_3)$
$\wedge$	$\mathsf{deleting}(w_3, z_3, w_4, x_4)$	$(d_4 = w_4 x_4)$
$\wedge$	$\mathsf{intersection}(w_3, x_3, w_4, z_3)$	
$\rightarrow$	$\mathsf{deleting}(y_2, z_3, w_4, x_3)$	$(d_3'(Y) = w_4 x_3(Step1/Step2))$
$\vee$	$\mathsf{deleting}(y_2, z_3, w_4, x_4)$	$(d_3'(Y) = w_4 x_4(Step1/Step2))$
$\vee$	$\mathsf{deleting}(w_3, z_3, x_4, w_4)$	$(d_{4b} = x_4 w_4(Step3a4))$

Step3a3:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, w_3, x_3)$	$(d_3(Y) = w_3 x_3)$
$\wedge\neg$	$\mathsf{F}(w_3, x_3)$	$(\leftarrow Step1b/2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, z_3)$	$(e_3 = w_3 z_3)$
$\wedge$	$\mathsf{deleting}(w_3, z_3, w_4, x_4)$	$(d_{4a} = w_4 x_4)$
$\wedge$	$\mathsf{deleting}(w_3, z_3, x_4, w_4)$	$(d_{4b} = x_4 w_4)$
$\rightarrow$	$\mathsf{deleting}(y_2, z_3, w_4, x_3)$	$(d'_3(Y) = w_4 x_3(Step1/Step2)$
$\vee$	$\mathsf{deleting}(y_2, z_3, w_4, x_4)$	$(d'_3(Y) = w_4 x_4(Step1/Step2))$
$\vee$	$\mathsf{deleting}(y_2, z_3, x_4, x_3)$	$(d'_3(Y) = x_4 x_3(Step1/Step2)$
$\vee$	$\mathsf{deleting}(y_2, z_3, x_4, w_4)$	$(d'_3(Y) = x_4 w_4(Step1/Step2))$
$\vee$	$\mathsf{inside}(y_2, w_3, z_3, w_4) \vee \mathsf{inside}(w_3, y_2, z_3, w_4)$	$(d_4(Y) : Step3b,$ $d_4(Z) : Step4a)$
$\vee$	$\mathsf{inside}(y_2, w_3, z_3, x_4) \vee \mathsf{inside}(w_3, y_2, z_3, x_4)$	$(d_4(Y) : Step3b,$ $d_4(Z) : Step4a)$

Step3a4:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, w_3, x_3)$	$(d_3(Y) = w_3 x_3)$
$\wedge\neg$	$\mathsf{F}(w_3, x_3)$	$(\leftarrow Step1b/2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, z_3)$	$(e_3 = w_3 z_3)$
$\wedge$	$\mathsf{deleting}(w_3, z_3, w_4, x_4)$	$(d_{4a} = w_4 x_4)$
$\wedge$	$\mathsf{intersection}(w_3, x_3, w_4, z_3)$	
$\wedge$	$\mathsf{deleting}(w_3, z_3, x_4, w_4)$	$(d_{4b} = x_4 w_4)$
$\rightarrow$	$\mathsf{deleting}(y_2, z_3, w_4, x_3)$	$(d'_3(Y) = w_4 x_3(Step1/Step2)$
$\vee$	$\mathsf{deleting}(y_2, z_3, w_4, x_4)$	$(d'_3(Y) = w_4 x_4(Step1/Step2))$
$\vee$	$\mathsf{deleting}(y_2, z_3, x_4, x_3)$	$(d'_3(Y) = x_4 x_3(Step1/Step2)$
$\vee$	$\mathsf{deleting}(y_2, z_3, x_4, w_4)$	$(d'_3(Y) = x_4 w_4(Step1/Step2))$

Step3a5:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, w_3, x_3)$	$(d_3(Y) = w_3 x_3)$
$\wedge\neg$	$\mathsf{F}(w_3, x_3)$	$(\leftarrow Step1b/2b)$
$\wedge$	$\mathsf{deleting}(w_3, z_3, w_4, x_4)$	$(d_{4a}(Y) = w_4 x_4)$
$\wedge$	$\mathsf{deleting}(w_3, z_3, x_4, w_4)$	$(d_{4b}(Y) = x_4 w_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_4, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, w_4)$	$(d_4 = d_{4a}, e_4 = w_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, x_4, z_3)$	$(d_4 = d_{4b}, e_4 = x_4 z_3)$
$\wedge$	$\mathsf{deleting}(w_4, z_3, w_5, x_5)$	$(d_4 = d_{4a}, d_5(Y) = w_5 x_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_5, z_3)$	$(d_4 = d_{4a}, e_5 = w_5 z_3)$
$\wedge$	$\mathsf{deleting}(w_5, z_3, w_4, x_4)$	$(d_4 = d_{4a}, d_6(Y) = w_4 x_4)$
$\rightarrow$	$\mathsf{deleting}(y_2, z_3, w_4, x_3)$	$(d_3'(Y) = w_4 x_3(Step1/Step2))$
$\vee$	$\mathsf{deleting}(y_2, z_3, x_4, x_3)$	$(d_3'(Y) = x_4 x_3(Step1/Step2))$
$\vee$	$\mathsf{deleting}(y_2, z_3, x_4, x_5)$	$(d_3'(Y) = x_4 x_5(Step1/Step2))$
$\vee$	$\mathsf{inside}(w_3, y_2, z_3, x_4)$	$(d_4 = d_{4b}(Step3b))$

Step3b1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, w_3, x_3)$	$(d_3(Y) = w_3 x_3)$
$\wedge\neg$	$\mathsf{F}(w_3, x_3)$	$(\leftarrow Step1b/2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, z_3)$	$(e_3 = w_3 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_3)$	$(e_4 = y_3 z_3)$
$\wedge\neg$	$\mathsf{intersection}(w_3, x_3, y_3, z_3)$	$(\leftarrow Step3a2)$
$\wedge$	$\mathsf{deleting}(y_3, z_3, w_5, x_5)$	$(d_5 = w_5 x_5)$
$\rightarrow$	$\mathsf{deleting}(y_2, z_3, w_5, x_3)$	$(d_3'(Y) = w_5 x_3(Step1/Step2))$
$\vee$	$\mathsf{deleting}(y_2, z_3, w_5, x_5)$	$(d_3'(Y) = w_5 x_5(Step1/Step2))$
$\vee$	$\mathsf{intersection}(w_3, x_3, w_5, z_3)$	$(Step3b2)$
$\vee$	$\mathsf{deleting}(y_3, z_3, x_5, w_5)$	$(d_{5b} = x_5 w_5(Step3b3))$
$\vee$	$\mathsf{inside}(w_3, y_2, z_3, w_5)$	$(d_5(Y) : Step3b\ with\ w_5\ as\ y_3,$ $d_5(Z) : Step4b)$

Step3b2:

$\mathsf{E}(u_1, v_1)$ $(e_0 = u_1v_1)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$ $(e_2 = y_2z_3)$

$\wedge$ $\mathsf{deleting}(y_2, z_3, w_3, x_3)$ $(d_3(Y) = w_3x_3)$

$\wedge\neg\mathsf{F}(w_3, x_3)$ $(\leftarrow Step1b/2b)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, w_3, z_3)$ $(e_3 = w_3z_3)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_3)$ $(e_4 = y_3z_3)$

$\wedge\neg\mathsf{intersection}(w_3, x_3, y_3, z_3)$ $(\leftarrow Step3a2)$

$\wedge$ $\mathsf{deleting}(y_3, z_3, w_5, x_5)$ $(d_5 = w_5x_5)$

$\wedge$ $\mathsf{intersection}(w_3, x_3, w_5, z_3)$

$\rightarrow$ $\mathsf{deleting}(y_2, z_3, w_5, x_3)$ $(d'_3(Y) = w_5x_3(Step1/Step2))$

$\vee$ $\mathsf{deleting}(y_2, z_3, w_5, x_5)$ $(d'_3(Y) = w_5x_5(Step1/Step2))$

$\vee$ $\mathsf{deleting}(y_3, z_3, x_5, w_5)$ $(d_{5b} = x_5w_5(Step3b3))$

Step3b3:

$\mathsf{E}(u_1, v_1)$ $(e_0 = u_1v_1)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$ $(e_2 = y_2z_3)$

$\wedge$ $\mathsf{deleting}(y_2, z_3, w_3, x_3)$ $(d_3(Y) = w_3x_3)$

$\wedge\neg\mathsf{F}(w_3, x_3)$ $(\leftarrow Step1b/2b)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, w_3, z_3)$ $(e_3 = w_3z_3)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_3)$ $(e_4 = y_3z_3)$

$\wedge\neg\mathsf{intersection}(w_3, x_3, y_3, z_3)$ $(\leftarrow Step3a2)$

$\wedge$ $\mathsf{deleting}(y_3, z_3, w_5, x_5)$ $(d_{5a} = w_5x_5)$

$\wedge$ $\mathsf{deleting}(y_3, z_3, x_5, w_5)$ $(d_{5b} = x_5w_5)$

$\rightarrow$ $\mathsf{deleting}(y_2, z_3, w_5, x_3)$ $(d'_3(Y) = w_5x_3(Step1/Step2))$

$\vee$ $\mathsf{deleting}(y_2, z_3, w_5, x_5)$ $(d'_3(Y) = w_5x_5(Step1/Step2))$

$\vee$ $\mathsf{deleting}(y_2, z_3, x_5, x_3)$ $(d'_3(Y) = x_5x_3(Step1/Step2))$

$\vee$ $\mathsf{deleting}(y_2, z_3, x_5, w_5)$ $(d'_3(Y) = x_5w_5(Step1/Step2))$

$\vee$ $\mathsf{intersection}(w_3, x_3, w_5, z_3)$ $(Step3b4)$

$\vee$ $\mathsf{inside}(w_3, y_2, z_3, w_5)$ $(d_5(Y) : Step3b\ with\ w_5\ as\ y_3,$
$d_5(Z) : Step4b)$

$\vee$ $\mathsf{inside}(w_3, y_2, z_3, x_5)$ $(d_5(Y) : Step3b\ with\ x_5\ as\ y_3,$
$d_5(Z) : Step4b)$

Step3b4:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3 x_3) \\
\wedge\neg & \mathsf{F}(w_3, x_3) & (\leftarrow Step1b/2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_3, z_3) \wedge \mathsf{E}(y_3, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_3) & (e_4 = y_3 z_3) \\
\wedge\neg & \mathsf{intersection}(w_3, x_3, y_3, z_3) & (\leftarrow Step3a2) \\
\wedge & \mathsf{deleting}(y_3, z_3, w_5, x_5) & (d_{5a} = w_5 x_5) \\
\wedge & \mathsf{deleting}(y_3, z_3, x_5, w_5) & (d_{5b} = x_5 w_5) \\
\wedge & \mathsf{intersection}(w_3, x_3, w_5, z_3) & \\
\rightarrow & \mathsf{deleting}(y_2, z_3, w_5, x_3) & (d'_3(Y) = w_5 x_3 (Step1/Step2)) \\
\vee & \mathsf{deleting}(y_2, z_3, w_5, x_5) & (d'_3(Y) = w_5 x_5 (Step1/Step2)) \\
\vee & \mathsf{deleting}(y_2, z_3, x_5, x_3) & (d'_3(Y) = x_5 x_3 (Step1/Step2)) \\
\vee & \mathsf{deleting}(y_2, z_3, x_5, w_5) & (d'_3(Y) = x_5 w_5 (Step1/Step2))
\end{array}
$$

Step3c1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, y_4, z_5) & (\leftarrow Step4a4/4b4\ for\ k-1) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge\neg & \mathsf{F}(w_5, x_5) & (\leftarrow Step2b) \\
\wedge & \mathsf{intersection}(y_2, z_3, w_5, z_5) & \\
\rightarrow \bot & &
\end{array}
$$

Step3c2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_5, z_5) \wedge \mathsf{E}(w_5, z_5) & (e_5 = w_5 z_5) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, w_5, z_5) & (\leftarrow Step2a2) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, w_5, z_5) & (\leftarrow Step2a3) \\
\wedge\neg & \mathsf{inside}(z_5, z_3, y_4, w_5) & (\leftarrow Step2c) \\
\wedge & \mathsf{inside}(w_5, y_4, z_5, y_5) & \\
\rightarrow & \bot &
\end{array}
$$

Step3c3:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_5, z_5) \wedge \mathsf{E}(w_5, z_5) & (e_5 = w_5 z_5) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, w_5, z_5) & (\leftarrow Step2a2) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, w_5, z_5) & (\leftarrow Step2a3) \\
\wedge\neg & \mathsf{inside}(z_5, z_3, y_4, w_5) & (\leftarrow Step2c) \\
\wedge & \mathsf{inside}(y_4, w_5, z_5, y_5) & \\
\wedge & \mathsf{intersection}(y_2, z_3, y_5, z_5) & \\
\rightarrow & \bot &
\end{array}
$$

Step3c4:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_5, z_5) \wedge \mathsf{E}(w_5, z_5) & (e_5 = w_5 z_5) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, w_5, z_5) & (\leftarrow Step2a2) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, w_5, z_5) & (\leftarrow Step2a3) \\
\wedge\neg & \mathsf{inside}(z_5, z_3, y_4, w_5) & (\leftarrow Step2c) \\
\wedge & \mathsf{inside}(y_4, w_5, z_5, y_5) & \\
\wedge & \mathsf{intersection}(y_4, z_3, y_5, z_5) & \\
\rightarrow & \bot &
\end{array}
$$

Step3d1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) & (e_5 = y_6 z_5) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step2a2\ for\ k-1) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, y_6, z_5) & (\leftarrow Step2a3\ for\ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) & (e_6 = y_6 z_7) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step4c2\ for\ k-1) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, y_6, z_7) & (\leftarrow Step4c3\ for\ k-1) \\
\wedge & \mathsf{deleting}(y_6, z_7, w_7, x_7) & (d_7(Y) = w_7 x_7) \\
\wedge\neg & \mathsf{F}(w_7, x_7) & (\leftarrow Step1b/2b) \\
\wedge & \mathsf{intersection}(y_2, z_3, w_7, z_7) & \\
\rightarrow & \bot &
\end{array}
$$

Step3d2:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) & (e_5 = y_6 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_7, z_7) & (e_7 = y_7 z_7) \\
\wedge & \neg\mathsf{intersection}(y_4, z_5, y_7, z_7) & (\leftarrow Step2a2/Step3c - f \ for \ j+1) \\
\wedge & \mathsf{intersection}(y_4, z_3, y_7, z_7) & \\
\rightarrow & \bot &
\end{array}
$$

Step3d3:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) & (e_6 = y_6 z_7) \\
\wedge & \mathsf{deleting}(y_6, z_7, w_7, x_7) & (d_7(Y) = w_7 x_7) \\
\wedge & \neg\mathsf{F}(w_7, x_7) & (\leftarrow Step2b) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step4c2 \ for \ k-1) \\
\wedge & \mathsf{intersection}(y_4, z_3, y_6, z_7) & (\leftarrow Step4c3 \ for \ k-1) \\
\wedge & \mathsf{inside}(w_7, y_6, z_7, y_2) & \\
\rightarrow & \bot &
\end{array}
$$

Step3d4:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) && (e_3 = y_4 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) && (e_6 = y_6 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_7) && (\leftarrow Step4d2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_6, z_7) && (\leftarrow Step4d3\ for\ k-1) \\
\wedge\ & \mathsf{deleting}(y_6, z_7, w_7, x_7) && (d_7(Y) = w_7 x_7) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, w_7, z_7) && (e_7 = w_7 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, w_7, z_7) && (\leftarrow Step3d1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, w_7, z_7) && (\leftarrow Step3d2) \\
\wedge & \neg \mathsf{inside}(w_7, y_6, z_7, y_2) && (\leftarrow Step3d3) \\
\wedge\ & \mathsf{inside}(w_7, y_6, z_7, y_7) \\
\wedge\ & \mathsf{intersection}(y_2, z_3, y_7, z_7) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step3e1:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) && (e_3 = y_4 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_3) && \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) && (e_6 = y_6 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_7) && (\leftarrow Step4c2\ for\ k-2) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_6, z_7) && (\leftarrow Step4c3\ for\ k-2) \\
\wedge\ & \mathsf{deleting}(y_6, z_7, y_8, x_7) && (d_7(Y) = y_8 x_7) \\
\wedge & \neg \mathsf{F}(y_8, x_7) && (\leftarrow Step1b/2b\ for\ k-1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_7) && (e_7 = y_8 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_7) && (\leftarrow Step2a2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_8, z_7) && (\leftarrow Step2a3\ for\ k-1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) && (e_8 = y_8 z_9) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_9) && (\leftarrow Step4c2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_8, z_9) && (\leftarrow Step4c3\ for\ k-1) \\
\wedge\ & \mathsf{deleting}(y_8, z_9, w_9, x_9) && (d_9(Y) = w_9 x_9) \\
\wedge & \neg \mathsf{F}(w_9, x_9) && (\leftarrow Step1b/2b) \\
\wedge\ & \mathsf{intersection}(y_2, z_3, w_9, z_9) && \\
\rightarrow\ & \bot &&
\end{aligned}
$$

Step3e2:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) && (e_3 = y_4 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_3) && \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) && (e_6 = y_6 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_7) && (\leftarrow Step4c2\ for\ k-2) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_6, z_7) && (\leftarrow Step4c3\ for\ k-2) \\
\wedge\ & \mathsf{deleting}(y_6, z_7, w_7, x_7) && (d_7(Y) = w_7 x_7) \\
\wedge & \neg \mathsf{F}(w_7, x_7) && (\leftarrow Step1b/2b\ for\ k-1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, w_7, z_7) && (e_7 = w_7 z_7) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(w_7, y_6, z_7, y_8) && (e_8 = y_8 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_7) && (\leftarrow Step2a2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_8, z_7) && (\leftarrow Step2a3\ for\ k-1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) && (e_9 = y_8 z_9) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_9) && (\leftarrow Step4c2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_8, z_9) && (\leftarrow Step4c3\ for\ k-1) \\
\wedge\ & \mathsf{deleting}(y_8, z_9, w_{10}, x_{10}) && (d_{10}(Y) = w_{10} x_{10}) \\
\wedge & \neg \mathsf{F}(w_{10}, x_{10}) && (\leftarrow Step1b/2b) \\
\wedge\ & \mathsf{intersection}(y_2, z_3, w_{10}, z_9) && \\
\rightarrow\ & \bot &&
\end{aligned}
$$

Step3e3:

$$
\begin{array}{lr}
\mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) & (e_5 = y_6 z_5) \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_8, z_5) & \\
\wedge\ \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_4, y_6, z_5, z_9) & (e_8 = y_8 z_9) \\
\wedge \neg \mathsf{intersection}(y_4, z_5, y_8, z_9) & (\leftarrow Step4c2\ for\ k-1) \\
\wedge \neg \mathsf{intersection}(y_6, z_5, y_8, z_9) & (\leftarrow Step4c3\ for\ k-1) \\
\wedge\ \mathsf{deleting}(y_8, z_9, w_9, x_9) & (d_9(Y) = w_9 x_9) \\
\wedge \neg \mathsf{F}(w_9, x_9) & (\leftarrow Step1b/Step2b) \\
\wedge\ \mathsf{inside}(y_8, w_9, z_9, y_4) & \\
\rightarrow\ \bot &
\end{array}
$$

Step3e4:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) \qquad\qquad (e_6 = y_6 z_7) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_6, z_7) \qquad\qquad (\leftarrow Step4c2 \ for \ k-2) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_6, z_7) \qquad\qquad (\leftarrow Step4c3 \ for \ k-2) \\
\wedge & \mathsf{deleting}(y_6, z_7, y_8, x_7) \qquad\qquad (d_7(Y) = y_8 x_7) \\
\wedge \neg & \mathsf{F}(y_8, x_7) \qquad\qquad (\leftarrow Step1b/2b \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \qquad\qquad (e_7 = y_8 z_7) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_8, z_7) \qquad\qquad (\leftarrow Step2a2 \ for \ k-1) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_8, z_7) \qquad\qquad (\leftarrow Step2a3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) \qquad\qquad (e_8 = y_8 z_9) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_8, z_9) \qquad\qquad (\leftarrow Step4c2 \ for \ k-1) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_8, z_9) \qquad\qquad (\leftarrow Step4c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_8, z_9, w_9, x_9) \qquad\qquad (d_9(Y) = w_9 x_9) \\
\wedge \neg & \mathsf{F}(w_9, x_9) \qquad\qquad (\leftarrow Step1b/2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_9, z_9) \qquad\qquad (e_9 = w_9 z_9) \\
\wedge & \mathsf{inside}(y_8, w_9, z_9, y_9) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_9, z_9) \\
\rightarrow & \ \bot
\end{array}
$$

Step3e5:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) & (e_6 = y_6 z_7) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step4c2 \ for \ k-2) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_6, z_7) & (\leftarrow Step4c3 \ for \ k-2) \\
\wedge & \mathsf{deleting}(y_6, z_7, w_7, x_7) & (d_7(Y) = w_7 x_7) \\
\wedge \neg & \mathsf{F}(w_7, x_7) & (\leftarrow Step1b/2b \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_7, z_7) & (e_7 = w_7 z_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(w_7, y_6, z_7, y_8) & (e_8 = y_8 z_7) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step2a2 \ for \ k-1) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_8, z_7) & (\leftarrow Step2a3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) & (e_9 = y_8 z_9) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_8, z_9) & (\leftarrow Step4c2 \ for \ k-1) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_8, z_9) & (\leftarrow Step4c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_8, z_9, w_{10}, x_{10}) & (d_{10}(Y) = w_{10} x_{10}) \\
\wedge \neg & \mathsf{F}(w_{10}, x_{10}) & (\leftarrow Step1b/2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_{10}, z_9) & (e_{10} = w_{10} z_9) \\
\wedge & \mathsf{inside}(y_8, w_{10}, z_9, y_9) & \\
\wedge & \mathsf{intersection}(y_2, z_3, y_9, z_9) & \\
\rightarrow & \bot &
\end{array}
$$

Step3f1:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(y_2, y_4) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) && (e_3 = y_4 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_3) && \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_4, y_2, z_3, z_9) && (e_8 = y_8 z_9) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_9) && (\leftarrow Step4c2\ for\ k-2) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_8, z_9) && (\leftarrow Step4c3\ for\ k-2) \\
\wedge\ & \mathsf{deleting}(y_8, z_9, y_{10}, x_9) && (d_9(Y) = y_{10} x_9) \\
\wedge & \neg \mathsf{F}(y_{10}, x_9) && (\leftarrow Step1b/2b\ for\ k-1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) && (e_9 = y_{10} z_9) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_{10}, z_9) && (\leftarrow Step2a2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_{10}, z_9) && (\leftarrow Step2a3\ for\ k-1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_{10}, z_{11}) \wedge \mathsf{E}(y_{10}, z_{11}) \wedge \mathsf{inside}(y_8, y_{10}, z_9, z_{11}) && (e_{10} = y_{10} z_{11}) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_{10}, z_{11}) && (\leftarrow Step4c2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_{10}, z_{11}) && (\leftarrow Step4c3\ for\ k-1) \\
\wedge\ & \mathsf{deleting}(y_{10}, z_{11}, w_{11}, x_{11}) && (d_{11}(Y) = w_{11} x_{11}) \\
\wedge & \neg \mathsf{F}(w_{11}, x_{11}) && (\leftarrow Step1b/2b) \\
\wedge\ & \mathsf{intersection}(y_2, z_3, w_{11}, z_{11}) && \\
\rightarrow\ & \bot &&
\end{aligned}
$$

Step3f2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_4, y_2, z_3, z_9) & (e_8 = y_8 z_9) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_8, z_9) & (\leftarrow Step4c2 \ for \ k-2) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_8, z_9) & (\leftarrow Step4c3 \ for \ k-2) \\
\wedge & \mathsf{deleting}(y_8, z_9, w_9, x_9) & (d_9(Y) = w_9 x_9) \\
\wedge & \neg\mathsf{F}(w_9, x_9) & (\leftarrow Step1b/2b \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_9, z_9) & (e_9 = w_9 z_9) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) \wedge \mathsf{inside}(y_8, w_9, z_9, y_{10}) & (e_{10} = y_{10} z_9) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_{10}, z_9) & (\leftarrow Step2a2 \ for \ k-1) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_{10}, z_9) & (\leftarrow Step2a3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_{11}) \wedge \mathsf{E}(y_{10}, z_{11}) \wedge \mathsf{inside}(y_8, y_{10}, z_9, z_{11}) & (e_{11} = y_{10} z_{11}) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_{10}, z_{11}) & (\leftarrow Step4c2 \ for \ k-1) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_{10}, z_{11}) & (\leftarrow Step4c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_{10}, z_{11}, w_{12}, x_{12}) & (d_{12}(Y) = w_{12} x_{12}) \\
\wedge & \neg\mathsf{F}(w_{12}, x_{12}) & (\leftarrow Step1b/2b) \\
\wedge & \mathsf{intersection}(y_2, z_3, w_{12}, z_{11}) & \\
\rightarrow & \bot &
\end{array}
$$

Step4a1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, y_4, x_3) & (d_3(Y) = y_4 x_3) \\
\wedge & \neg\mathsf{F}(y_4, x_3) & (Step1b/2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) & (d_4(Z) = w_4 x_4) \\
\rightarrow & \mathsf{deleting}(y_4, z_3, x_4, w_4) & (d_{4b} = x_4 w_4 (Step4a3)) \\
\vee & \mathsf{inside}(y_2, y_4, z_3, w_4) & (Step4b) \\
\vee & \mathsf{inside}(y_4, y_2, z_3, w_4) & (Step4b)
\end{array}
$$

Step4a2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, w_3, x_3)$	$(d_3(Y) = w_3 x_3)$
$\wedge\neg$	$\mathsf{F}(w_3, x_3)$	$(\leftarrow Step1b/Step2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_4)$	$(e_4 = y_4 z_3)$
$\wedge\neg$	$\mathsf{intersection}(w_3, x_3, y_4, z_3)$	$(\leftarrow Step3a2/3b2/3b4)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_5, x_5)$	$(d_5(Z) = w_5 x_5)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, x_5, w_5)$	$(d_{5b} = x_5 w_5 (Step4a3))$
$\vee$	$\mathsf{inside}(y_4, y_2, z_3, w_5)$	$(Step4b)$

Step4a3:

	$\mathsf{E}(u_1, v_1) \wedge \neg\mathsf{F}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_{4a}(Z) = w_4 x_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, x_4, w_4)$	$(d_{4b}(Z) = x_4 w_4)$
$\rightarrow$	$\mathsf{inside}(y_2, y_4, z_3, w_4) \vee \mathsf{inside}(y_4, y_2, z_3, w_4)$	$(Step4b)$
$\vee$	$\mathsf{inside}(y_2, y_4, z_3, x_4) \vee \mathsf{inside}(y_4, y_2, z_3, x_4)$	$(Step4b)$

Step4a4:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, y_4, x_3)$	$(d_3(Y) = y_4 x_3)$
$\wedge\neg$	$\mathsf{F}(y_4, x_3)$	$(Step1b/Step2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_{4a}(Z) = w_4 x_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, x_4, w_4)$	$(d_{4b}(Z) = x_4 w_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4) \wedge \mathsf{inside}(y_4, y_2, z_3, w_4)$	$(d_4 = d_{4a}, e_4 = y_4 w_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, x_4)$	$(d_4 = d_{4b}, e_4 = y_4 x_4)$
$\wedge$	$\mathsf{deleting}(y_4, w_4, w_5, x_5)$	$(d_4 = d_{4a}, d_5(Z) = w_5 x_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_5)$	$(d_4 = d_{4a}, e_5 = y_4 w_5)$
$\wedge$	$\mathsf{deleting}(y_4, w_5, w_4, x_4)$	$(d_4 = d_{4a}, d_6(Z) = w_4 x_4)$
$\rightarrow$	$\mathsf{inside}(y_4, y_2, z_3, x_4)$	$(d_4 = d_{4b}(Step4b))$

Step4a5:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) \qquad\qquad (d_3(Y) = w_3 x_3) \\
\wedge & \neg\mathsf{F}(w_3, x_3) \qquad\qquad (\leftarrow Step1b/Step2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_4) \qquad\qquad (e_4 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_5, x_5) \qquad\qquad (d_{5a}(Z) = w_5 x_5) \\
\wedge & \mathsf{deleting}(y_4, z_3, x_5, w_5) \qquad\qquad (d_{5b}(Z) = x_5 w_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_5) \wedge \mathsf{inside}(y_4, y_2, z_3, w_5) \qquad\qquad (d_5 = d_{5a}, e_5 = y_4 w_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, x_5) \qquad\qquad (d_5 = d_{5b}, e_5 = y_4 x_5) \\
\wedge & \mathsf{deleting}(y_4, w_5, w_6, x_6) \qquad\qquad (d_5 = d_{5a}, d_6(Z) = w_6 x_6) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_6) \qquad\qquad (d_5 = d_{5a}, e_6 = y_4 w_6) \\
\wedge & \mathsf{deleting}(y_4, w_6, w_5, x_5) \qquad\qquad (d_5 = d_{5a}, d_7(Z) = w_5 x_5) \\
\rightarrow & \mathsf{inside}(y_4, y_2, z_3, x_5) \qquad\qquad (d_5 = d_{5b}(Step4b))
\end{array}
$$

Step4a6:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \wedge \neg\mathsf{F}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) \qquad\qquad (d_4(Z) = w_4 x_4) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_4, w_4) \\
\rightarrow & \bot
\end{array}
$$

Step4b1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, y_4, x_3)$	$(d_3(Y) = y_4 x_3)$
$\wedge\neg$	$\mathsf{F}(y_4, z_3)$	$(Step1b/Step2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_4, y_2, z_3, z_4)$	$(e_4 = y_4 z_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_5, x_5)$	$(d_5 = w_5 x_5)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_4, x_5, w_5)$	$(d_{5b} = x_5 w_5 (Step4b3))$
$\vee$	$\mathsf{inside}(y_4, y_2, z_3, w_5)$	$(d_5(Z) : Step4b\ with\ w_5\ as\ z_4,$
		$d_5(Y) : Step2\ for\ k+1)$
$\vee$	$\mathsf{intersection}(y_4, y_2, w_5, z_4)$	$(d_5(Y) Step3 for\ k+1)$

Step4b2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, w_3, x_3)$	$(d_3(Y) = w_3 x_3)$
$\wedge\neg$	$\mathsf{F}(w_3, x_3)$	$(Step1b/Step2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_3, z_3)$	$(e_3 = w_3 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_4)$	$(e_4 = y_4 z_3)$
$\wedge\neg$	$\mathsf{intersection}(w_3, x_3, y_4, z_3)$	$(\leftarrow Step3a2/3b2/3b4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_4, y_2, z_3, z_4)$	$(e_5 = y_4 z_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_6, x_6)$	$(d_6 = w_6 x_6)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_4, x_6, w_6)$	$(d_{6b} = x_6 w_6 (Step4b3))$
$\vee$	$\mathsf{inside}(y_4, y_2, z_3, w_6)$	$(d_6(Z) : Step4b\ with\ w_6\ as\ z_4,$
		$d_6(Y) : Step2\ for\ k+1)$
$\vee$	$\mathsf{intersection}(y_4, y_2, w_6, z_4)$	$(d_6(Y) : Step3 for\ k+1)$

Step4b3:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_4, y_2, z_3, z_4)$	$(e_4 = y_3 z_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_5, x_5)$	$(d_{5a} = w_5 x_5)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, x_5, w_5)$	$(d_{5b} = x_5 w_5)$
$\rightarrow$	$\mathsf{inside}(y_4, y_2, z_3, w_5)$	$(d_5(Z) : Step4b\ with\ w_5\ as\ z_4,$ $d_5(Y) : Step2\ for\ k+1)$
$\vee$	$\mathsf{inside}(y_4, y_2, z_3, x_5)$	$(d_5(Z) : Step4b\ with\ x_5\ as\ z_4,$ $d_5(Y) : Step2\ for\ k+1)$

Step4b4:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{deleting}(y_2, z_3, y_4, x_3)$	$(d_3(Y) = y_4 x_3)$
$\wedge$	$\neg\mathsf{F}(y_4, x_3)$	$(\leftarrow Step1b/Step2b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_4, y_2, z_3, z_4)$	$(e_4 = y_4 z_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_5, x_5)$	$(d_5(Z) = w_5 x_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_5) \wedge \mathsf{inside}(y_4, y_2, z_3, w_5)$	$(e_5 = y_4 w_5)$
$\wedge$	$\mathsf{intersection}(y_2, z_3, y_4, w_5)$	
$\rightarrow$	$\perp$	

Step4b5:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3 x_3) \\
\wedge & \neg \mathsf{F}(w_3, x_3) & (\leftarrow Step1b/Step2b) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_4) & (e_4 = y_4 z_3) \\
\wedge & \neg \mathsf{intersection}(w_3, x_3, y_4, z_3) & (\leftarrow Step3b2/3b4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_4, y_2, z_3, z_4) & (e_5 = y_4 z_4) \\
\wedge & \mathsf{deleting}(y_4, z_4, w_6, x_6) & (d_6(Z) = w_6 x_6) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_6) \wedge \mathsf{inside}(y_4, y_2, z_3, w_6) & (e_6 = y_4 w_6) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_4, w_6) & \\
\rightarrow & \bot &
\end{array}
$$

Step4c1:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) & (e_5 = y_6 z_5) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step2a2) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_6, z_5) & (\leftarrow Step2a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) & (e_6 = y_6 z_7) \\
\rightarrow & \mathsf{inside}(y_4, y_2, z_3, z_7) &
\end{array}
$$

Step4c2:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) && (e_3 = y_4 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) && (e_4 = y_4 z_5) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) && (e_5 = y_6 z_5) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_5) && (\leftarrow Step2a2) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_6, z_5) && (\leftarrow Step2a3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) && (e_6 = y_6 z_7) \\
\wedge\ & \mathsf{intersection}(y_2, z_3, y_6, z_7) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step4c3:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) && (e_3 = y_4 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) && (e_4 = y_4 z_5) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) && (e_5 = y_6 z_5) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_5) && (\leftarrow Step2a2) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_6, z_5) && (\leftarrow Step2a3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) && (e_6 = y_6 z_7) \\
\wedge\ & \mathsf{intersection}(y_4, z_3, y_6, z_7) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step4d1:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) & (e_6 = y_6 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step4c2/e2 \ for \ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_6, z_7) & (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) & (e_7 = y_8 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step2a2) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_8, z_7) & (\leftarrow Step2a3) \\
\wedge & \neg \mathsf{inside}(y_8, y_6, z_7, y_2) & (\leftarrow Step3c3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) & (e_8 = y_8 z_9) \\
\rightarrow & \mathsf{inside}(y_4, y_2, z_3, z_9) &
\end{array}
$$

Step4d2:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) & (e_6 = y_6 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step4c2/e2 \ for \ k-1) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_6, z_7) & (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) & (e_7 = y_8 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step2a2) \\
\wedge & \neg \mathsf{intersection}(y_4, z_3, y_8, z_7) & (\leftarrow Step2a3) \\
\wedge & \neg \mathsf{inside}(y_8, y_6, z_7, y_2) & (\leftarrow Step3c3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) & (e_8 = y_8 z_9) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_8, z_9) & \\
\rightarrow & \bot &
\end{array}
$$

Step4d3:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) & (e_6 = y_6 z_7) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step4c2/e2 \ for \ k-1) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_6, z_7) & (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) & (e_7 = y_8 z_7) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step2a2) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_8, z_7) & (\leftarrow Step2a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) & (e_8 = y_8 z_9) \\
\wedge & \mathsf{intersection}(y_4, z_3, y_8, z_9) & \\
\rightarrow & \bot &
\end{array}
$$

Step4e1:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_4, y_2, z_3, z_9) & (e_8 = y_8 z_9) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_8, z_9) & (\leftarrow Step4c2/e2 \ for \ k-1) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_8, z_9) & (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) & (e_9 = y_{10} z_9) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_{10}, z_9) & (\leftarrow Step2a2) \\
\wedge \neg & \mathsf{intersection}(y_4, z_3, y_{10}, z_9) & (\leftarrow Step2a3) \\
\wedge \neg & \mathsf{inside}(y_8, y_{10}, z_9, y_4) & (\leftarrow Step3c3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_{11}) \wedge \mathsf{E}(y_{10}, z_{11}) \wedge \mathsf{inside}(y_8, y_{10}, z_9, z_{11}) & (e_{10} = y_{10} z_{11}) \\
\rightarrow & \mathsf{inside}(y_4, y_2, z_3, z_{11}) &
\end{array}
$$

Step4e2:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_4, y_2, z_3, z_9) & (e_8 = y_8 z_9) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_8, z_9) & (\leftarrow Step4c2/e2 \ for \ k-1) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_8, z_9) & (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) & (e_9 = y_{10} z_9) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_{10}, z_9) & (\leftarrow Step2a2) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_{10}, z_9) & (\leftarrow Step2a3) \\
\wedge & \neg\mathsf{inside}(y_8, y_{10}, z_9, y_4) & (\leftarrow Step3c3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_{11}) \wedge \mathsf{E}(y_{10}, z_{11}) \wedge \mathsf{inside}(y_8, y_{10}, z_9, z_{11}) & (e_{10} = y_{10} z_{11}) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_{10}, z_{11}) & \\
\rightarrow & \bot &
\end{array}
$$

Step4e3:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_3) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_4, y_2, z_3, z_9) & (e_8 = y_8 z_9) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_8, z_9) & (\leftarrow Step4c2/e2 \ for \ k-1) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_8, z_9) & (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) & (e_9 = y_{10} z_9) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_{10}, z_9) & (\leftarrow Step2a2) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_{10}, z_9) & (\leftarrow Step2a3) \\
\wedge & \neg\mathsf{inside}(y_8, y_{10}, z_9, y_4) & (\leftarrow Step3c3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_{11}) \wedge \mathsf{E}(y_{10}, z_{11}) \wedge \mathsf{inside}(y_8, y_{10}, z_9, z_{11}) & (e_{10} = y_{10} z_{11}) \\
\wedge & \mathsf{intersection}(y_4, z_3, y_{10}, z_{11}) & \\
\rightarrow & \bot &
\end{array}
$$

Step5a:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) & (d_5(Y) = w_5 x_5) \\
\rightarrow & \mathsf{F}(w_5, x_5) & (\rightarrow \mathit{Step2b}) \\
\vee & \mathsf{inside}(z_3, z_1, y_2, w_5) &
\end{array}
$$

Step5b:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_4, y_2, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_3, z_1, y_2, y_6) & (e_5 = y_6 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_6, z_5, z_7) & (e_6 = y_6 z_7) \\
\wedge & \mathsf{deleting}(y_6, z_7, w_7, x_7) & (d_7(Y) = w_7 x_7) \\
\rightarrow & \mathsf{F}(w_7, x_7) & (\rightarrow \mathit{Step2b}) \\
\vee & \mathsf{inside}(z_3, z_1, y_2, w_7) &
\end{array}
$$

Step5c1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) \qquad\qquad (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_6) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) \qquad\qquad (e_6 = y_6 z_7) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, y_6, z_7) \qquad\qquad (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_6, z_7, y_8, x_7) \qquad\qquad (d_7(Y) = y_8 x_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{inside}(z_3, z_1, y_2, y_8) \qquad\qquad (e_7 = y_8 z_7) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, y_8, z_7) \qquad\qquad (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) \qquad\qquad (e_8 = y_8 z_9) \\
\wedge & \mathsf{deleting}(y_8, z_9, w_9, x_9) \qquad\qquad (d_9(Y) = w_9 x_9) \\
\rightarrow & \mathsf{F}(w_9, x_9) \qquad\qquad (\rightarrow Step2b) \\
\vee & \mathsf{inside}(z_3, z_1, y_2, w_9)
\end{array}
$$

Step5c2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) \qquad\qquad (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_6) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_4, y_2, z_3, z_7) \qquad\qquad (e_6 = y_6 z_7) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, y_6, z_7) \qquad\qquad (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_6, z_7, w_7, x_7) \qquad\qquad (d_7(Y) = w_7 x_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_7, z_7) \wedge \mathsf{inside}(z_3, z_1, y_2, w_7) \qquad\qquad (e_7 = w_7 z_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(w_7, y_6, z_7, y_8) \qquad\qquad (e_8 = y_8 z_7) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, y_8, z_7) \qquad\qquad (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_8, y_6, z_7, z_9) \qquad\qquad (e_9 = y_8 z_9) \\
\wedge & \mathsf{deleting}(y_8, z_9, w_{10}, x_{10}) \qquad\qquad (d_{10}(Y) = w_{10} x_{10}) \\
\rightarrow & \mathsf{F}(w_{10}, x_{10}) \qquad\qquad (\rightarrow Step2b) \\
\vee & \mathsf{inside}(z_3, z_1, y_2, w_{10})
\end{array}
$$

Step5d:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) \qquad\qquad (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_3) \wedge \mathsf{inside}(z_3, z_1, y_2, y_8) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(y_4, y_2, z_3, z_9) \qquad\qquad (e_8 = y_8 z_9) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_8, z_9) \qquad\qquad (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) \wedge \mathsf{inside}(z_3, z_1, y_2, y_{10}) \qquad\qquad (e_9 = y_{10} z_9) \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_{10}, z_9) \qquad\qquad (\leftarrow Step4c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_{11}) \wedge \mathsf{E}(y_{10}, z_{11}) \wedge \mathsf{inside}(y_8, y_{10}, z_9, z_{11}) \qquad\qquad (e_{10} = y_{10} z_{11}) \\
\wedge & \mathsf{deleting}(y_{10}, z_{11}, w_{11}, x_{11}) \qquad\qquad (d_{11}(Y) = w_{11} x_{11}) \\
\rightarrow & \mathsf{F}(w_{11}, x_{11}) \qquad\qquad (\rightarrow Step2b) \\
\vee & \mathsf{inside}(z_3, z_1, y_2, w_{11})
\end{array}
$$

Step6a1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) \qquad\qquad (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) \qquad\qquad (d_4(Z) = w_4 x_4) \\
\rightarrow & \mathsf{F}(w_4, x_4) \qquad\qquad (Step6b1) \\
\vee & \mathsf{inside}(y_2, y_4, z_3, w_4) \qquad\qquad (Step6c) \\
\vee & \mathsf{inside}(z_1, z_3, y_4, w_4) \qquad\qquad (Step7) \\
\vee & \mathsf{intersection}(z_1, z_3, y_4, w_4) \qquad\qquad (Step7)
\end{array}
$$

Step6a2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \wedge \neg\mathsf{F}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) \qquad\qquad (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) \qquad\qquad (d_4(Z) = w_4 x_4) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_4, w_4) \\
\rightarrow & \bot
\end{array}
$$

Step6b1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1)$	$(e_1 = y_2 z_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge$	$\mathsf{F}(w_4, x_4)$	
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, x_4, w_4)$	$(d_{4b}(Z) = x_4 w_4(Step6b2))$
$\vee$	$\mathsf{inside}(y_2, y_4, z_3, w_4)$	$(Step6c)$

Step6b2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1)$	$(e_1 = y_2 z_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_{4a}(Z) = w_4 x_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, x_4, w_4)$	$(d_{4b}(Z) = x_4 w_4)$
$\rightarrow$	$\mathsf{inside}(y_2, y_4, z_3, w_4)$	$(Step6c)$
$\vee$	$\mathsf{inside}(y_2, y_4, z_3, x_4)$	$(Step6c)$

Step6c1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4)$	$(e_4 = y_4 z_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_5, x_5)$	$(d_5 = w_5 x_5)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_5, x_5)$	$(d'_4(Z) = w_5 x_5(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_4, x_5, w_5)$	$(d_{5b} = x_5 w_5(Step6d))$
$\vee$	$\mathsf{deleting}(y_4, z_4, z_3, x_5)$	$(d'_5(Z) = z_3 x_5(Step6e))$
$\vee$	$\mathsf{inside}(y_2, y_4, z_3, w_5)$	$(d_5(Z) : Step2c\ with\ w_5\ as\ z_4,$ $d_5(Y) : Step6)$
$\vee$	$\mathsf{intersection}(u_1, v_1, w_5, z_4)$	$(d_5(Y) : Step2\ for\ k+1)$

Step6c2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4)$	$(e_4 = y_4 z_4)$
$\wedge\neg$	$\mathsf{intersection}(y_2, z_3, y_4, z_4)$	$(\leftarrow Step6a3)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_5, x_5)$	$(d_5(Z) = w_5 x_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_5)$	$(e_5 = y_4 w_5)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_4, z_3, x_5)$	$(d'_5(Z) = z_3 x_5 (Step6e))$

Step6d1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4)$	$(e_4 = y_4 z_4)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_5, x_5)$	$(d_{5a} = w_5 x_5)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, x_5, w_5)$	$(d_{5b} = x_5 w_5)$
$\rightarrow$	$\mathsf{inside}(y_2, y_4, z_3, w_5)$	$(d_5(Z) : Step6c\ with\ w_5\ as\ z_4,$ $d_5(Y) : Step2\ for\ k+1)$
$\vee$	$\mathsf{inside}(y_2, y_4, z_3, x_5)$	$(d_5(Z) : Step6c\ with\ x_5\ as\ z_4,$ $d_5(Y) : Step2\ for\ k+1)$

Step6d2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) \qquad\qquad (d_{4a}(Z) = w_4 x_4) \\
\wedge & \mathsf{deleting}(y_4, z_3, x_4, w_4) \qquad\qquad (d_{4b}(Z) = x_4 w_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_4) \wedge \mathsf{inside}(y_2, y_4, z_3, w_4) \qquad\qquad (d_4 = d_{4a}, e_4 = y_4 w_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, x_4) \qquad\qquad (d_4 = d_{4b}, e_4 = y_4 x_4) \\
\wedge & \mathsf{deleting}(y_4, w_4, w_5, x_5) \qquad\qquad (d_4 = d_{4a}, d_5(Z) = w_5 x_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_5) \qquad\qquad (d_4 = d_{4a}, e_5 = y_4 w_5) \\
\wedge & \mathsf{deleting}(y_4, w_5, w_4, x_4) \qquad\qquad (d_4 = d_{4a}, d_6(Z) = w_4 x_4) \\
\rightarrow & \mathsf{deleting}(y_4, z_3, x_4, x_5) \qquad\qquad (d'_4(Z) = x_4 x_5 (Step6)) \\
\vee & \mathsf{deleting}(y_4, w_4, z_3, x_5) \qquad\qquad (d_4 = d_{4a}, d'_5(Z) = z_3 x_5 (Step6e))
\end{array}
$$

Step6d3:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4) \qquad\qquad (e_4 = y_4 z_4) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, y_4, z_4) \qquad\qquad (\leftarrow Step6a3) \\
\wedge & \mathsf{deleting}(y_4, z_4, w_5, x_5) \qquad\qquad (d_{5a}(Z) = w_5 x_5) \\
\wedge & \mathsf{deleting}(y_4, z_4, x_5, w_5) \qquad\qquad (d_{5b}(Z) = x_5 w_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_5) \wedge \mathsf{inside}(y_2, y_4, z_3, w_5) \qquad\qquad (d_5 = d_{5a}, e_5 = y_4 w_5) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_4, w_5) \\
\rightarrow & \bot
\end{array}
$$

Step6e1:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, y_4, x_3) & (d_3(Y) = y_4 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4) & (e_4 = y_4 z_4) \\
\wedge & \mathsf{deleting}(y_4, z_4, z_3, x_5) & (d_5(Z) = z_3 x_5) \\
\wedge & \neg\mathsf{deleting}(y_4, z_4, x_5, z_3) & (\leftarrow Step6d) \\
\rightarrow & \bot
\end{array}
$$

Step6e2:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(y_2, w_3, z_3, y_4) & (e_4 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4) & (e_5 = y_4 z_4) \\
\wedge & \mathsf{deleting}(y_4, z_4, z_3, x_6) & (d_6(Z) = z_3 x_6) \\
\wedge & \neg\mathsf{deleting}(y_4, z_4, x_6, z_3) & (\leftarrow Step6d) \\
\rightarrow & \bot
\end{array}
$$

Step6e3:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(w_3, y_2, z_3, y_4) & (e_4 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4) & (e_5 = y_4 z_4) \\
\wedge & \mathsf{deleting}(y_4, z_4, z_3, x_6) & (d_6(Z) = z_3 x_6) \\
\wedge & \neg\mathsf{deleting}(y_4, z_4, x_6, z_3) & (\leftarrow Step6d) \\
\rightarrow & \bot
\end{array}
$$

Step6e4:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3 z_3) \\
\wedge & \mathsf{deleting}(w_3, z_3, y_4, x_4) & (d_4(Y) = y_4 x_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) & (e_4 = y_4 z_3) \\
\wedge & \neg\mathsf{deleting}(w_3, z_3, x_4, y_4) & (\leftarrow Part2, Step6) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4) & (e_5 = y_4 z_4) \\
\wedge & \mathsf{deleting}(y_4, z_4, z_3, x_6) & (d_6(Z) = z_3 x_6) \\
\wedge & \neg\mathsf{deleting}(y_4, z_4, x_6, z_3) & (\leftarrow Step6d) \\
\rightarrow & \bot
\end{array}
$$

Step6e5:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) & (d_3(Y) = w_3 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) & (e_3 = w_3 z_3) \\
\wedge & \mathsf{deleting}(w_3, z_3, w_4, x_4) & (d_4(Y) = w_4 x_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_4, z_3) & (e_4 = w_4 z_3) \\
\wedge & \neg\mathsf{deleting}(w_3, z_3, x_4, w_4) & (\leftarrow Part2, Step3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_3, w_3, w_4, y_4) & (e_5 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4) & (e_6 = y_4 z_4) \\
\wedge & \mathsf{deleting}(y_4, z_4, z_3, x_7) & (d_7(Z) = z_3 x_7) \\
\wedge & \neg\mathsf{deleting}(y_4, z_4, x_7, z_3) & (\leftarrow Step6d) \\
\rightarrow & \bot
\end{array}
$$

Step6e6:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{deleting}(y_2, z_3, w_3, x_3) \qquad\qquad (d_3(Y) = w_3 x_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_3, z_3) \qquad\qquad (e_3 = w_3 z_3) \\
\wedge & \mathsf{deleting}(w_3, z_3, w_4, x_4) \qquad\qquad (d_4(Y) = w_4 x_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_4, z_3) \qquad\qquad (e_4 = w_4 z_3) \\
\wedge\neg & \mathsf{deleting}(w_3, z_3, x_4, w_4) \qquad\qquad (\leftarrow Part2, Step3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_3, w_4, w_3, y_4) \qquad\qquad (e_5 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(y_2, y_4, z_3, z_4) \qquad\qquad (e_6 = y_4 z_4) \\
\wedge & \mathsf{deleting}(y_4, z_4, z_3, x_7) \qquad\qquad (d_7(Z) = z_3 x_7) \\
\wedge\neg & \mathsf{deleting}(y_4, z_4, x_7, z_3) \qquad\qquad (\leftarrow Step6d) \\
\rightarrow & \bot
\end{array}
$$

Step7a1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) \qquad\qquad (d_4(Z) = w_4 x_4) \\
\wedge\neg & \mathsf{F}(w_4, x_4) \qquad\qquad (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_4) \qquad\qquad (e_4 = y_4 w_4) \\
\wedge & \mathsf{deleting}(y_4, w_4, w_5, x_5) \qquad\qquad (d_5 = w_5 x_5) \\
\rightarrow & \mathsf{deleting}(y_4, z_3, w_5, x_4) \qquad\qquad (d'_4(Z) = w_5 x_4 (Step6)) \\
\vee & \mathsf{deleting}(y_4, z_3, w_5, x_5) \qquad\qquad (d'_4(Z) = w_5 x_5 (Step6)) \\
\vee & \mathsf{intersection}(w_4, x_4, y_4, w_5) \qquad\qquad (Step7a2) \\
\vee & \mathsf{deleting}(y_4, w_4, x_5, w_5) \qquad\qquad (d_{5b} = x_5 w_5 (Step7a3)) \\
\vee & \mathsf{inside}(z_3, w_4, y_4, w_5) \vee \mathsf{inside}(w_4, z_3, y_4, w_5) \qquad\qquad (d_5(Z) : Step7b, \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad d_5(Y) : Step8a)
\end{array}
$$

Step7a2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_2 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4)$	$(e_4 = y_4 w_4)$
$\wedge$	$\mathsf{deleting}(y_4, w_4, w_5, x_5)$	$(d_5 = w_5 x_5)$
$\wedge$	$\mathsf{intersection}(w_4, x_4, y_4, w_5)$	
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_5, x_4)$	$(d'_4(Z) = w_5 x_4 (Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, w_5, x_5)$	$(d'_4(Z) = w_5 x_5 (Step6))$
$\vee$	$\mathsf{deleting}(y_4, w_4, x_5, w_5)$	$(d_{5b} = x_5 w_5 (Step7a4))$

Step7a3:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4)$	$(e_4 = y_4 w_4)$
$\wedge$	$\mathsf{deleting}(y_4, w_4, w_5, x_5)$	$(d_{5a} = w_5 x_5)$
$\wedge$	$\mathsf{deleting}(y_4, w_4, x_5, w_5)$	$(d_{5b} = x_5 w_5)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_5, x_4)$	$(d'_4(Z) = w_5 x_4 (Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, w_5, x_5)$	$(d'_4(Z) = w_5 x_5 (Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_5, x_4)$	$(d'_4(Z) = x_5 x_4 (Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_5, w_5)$	$(d'_4(Z) = x_5 w_5 (Step6))$
$\vee$	$\mathsf{inside}(z_3, w_4, y_4, w_5) \vee \mathsf{inside}(w_4, z_3, y_4, w_5)$	$(d_5(Z) : Step7b,$ $d_5(Y) : Step8a)$
$\vee$	$\mathsf{inside}(z_3, w_4, y_4, x_5) \vee \mathsf{inside}(w_4, z_3, y_4, x_5)$	$(d_5(Z) : Step7b,$ $d_5(Y) : Step8a)$

Step7a4:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_2 = y_4z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4)$	$(e_4 = y_4w_4)$
$\wedge$	$\mathsf{deleting}(y_4, w_4, w_5, x_5)$	$(d_{5a} = w_5x_5)$
$\wedge$	$\mathsf{intersection}(w_4, x_4, y_4, w_5)$	
$\wedge$	$\mathsf{deleting}(y_4, w_4, x_5, w_5)$	$(d_{5b} = x_5w_5)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_5, x_4)$	$(d'_4(Z) = w_5x_4(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, w_5, x_5)$	$(d'_4(Z) = w_5x_5(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_5, x_4)$	$(d'_4(Z) = x_5x_4(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_5, w_5)$	$(d'_4(Z) = x_5w_5(Step6))$

Step7a5:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{deleting}(y_4, w_4, w_5, x_5)$	$(d_{5a}(Z) = w_5x_5)$
$\wedge$	$\mathsf{deleting}(y_4, w_4, x_5, w_5)$	$(d_{5b}(Z) = x_5w_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_5) \wedge \mathsf{inside}(z_3, w_4, y_4, w_5)$	$(d_5 = d_{5a}, e_5 = y_4w_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, x_5)$	$(d_5 = d_{5b}, e_5 = y_4x_5)$
$\wedge$	$\mathsf{deleting}(y_4, w_5, w_6, x_6)$	$(d_5 = d_{5a}, d_6(Z) = w_6x_6)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_6)$	$(d_5 = d_{5a}, e_6 = y_4w_6)$
$\wedge$	$\mathsf{deleting}(y_4, w_6, w_5, x_5)$	$(d_5 = d_{5a}, d_7(Z) = w_5x_5)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_5, x_4)$	$(d'_4(Z) = w_5x_4(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_5, x_4)$	$(d'_4(Z) = x_5x_4(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_5, x_6)$	$(d'_4(Z) = x_5x_6(Step6))$
$\vee$	$\mathsf{inside}(z_3, w_4, y_4, x_5)$	$(d_5 = d_{5b}(Step7b))$

Step7b1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4)$	$(e_4 = y_4 w_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(z_3, w_4, y_4, z_4)$	$(e_5 = y_4 z_4)$
$\wedge\neg$	$\mathsf{intersection}(w_4, x_4, y_4, z_4)$	$(\leftarrow Step7a2)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_6, x_6)$	$(d_6 = w_6 x_6)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_6, x_4)$	$(d'_4(Z) = w_6 x_4(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, w_6, x_6)$	$(d'_4(Z) = w_6 x_6(Step6))$
$\vee$	$\mathsf{intersection}(w_4, x_4, y_4, w_6)$	$(Step7b2)$
$\vee$	$\mathsf{deleting}(y_4, z_4, x_6, w_6)$	$(d_{6b} = x_6 w_6(Step7b3))$
$\vee$	$\mathsf{inside}(z_3, w_4, y_4, w_6)$	$(d_6(Z) : Step7b\ with\ w_6\ as\ z_4,$ $d_6(Y) : Step8b)$

Step7b2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4)$	$(e_4 = y_4 w_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(z_3, w_4, y_4, z_4)$	$(e_5 = y_4 z_4)$
$\wedge\neg$	$\mathsf{intersection}(w_4, x_4, y_4, z_4)$	$(\leftarrow Step7a2)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_6, x_6)$	$(d_6 = w_6 x_6)$
$\wedge$	$\mathsf{intersection}(w_4, x_4, y_4, w_6)$	
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_6, x_4)$	$(d'_4(Z) = w_6 x_4(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, w_6, x_6)$	$(d'_4(Z) = w_6 x_6(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_4, x_6, w_6)$	$(d_{6b} = x_6 w_6(Step7b3))$

Step7b3:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4)$	$(e_4 = y_4 w_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(z_3, w_4, y_4, z_4)$	$(e_5 = y_4 z_4)$
$\wedge\neg$	$\mathsf{intersection}(w_4, x_4, y_4, z_4)$	$(\leftarrow Step7a2)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_6, x_6)$	$(d_{6a} = w_6 x_6)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, x_6, w_6)$	$(d_{6b} = x_6 w_6)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_6, x_4)$	$(d'_4(Z) = w_6 x_4(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, w_6, x_6)$	$(d'_4(Z) = w_6 x_6(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_6, x_4)$	$(d'_4(Z) = x_6 x_4(Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_6, w_6)$	$(d'_4(Z) = x_6 w_6(Step6))$
$\vee$	$\mathsf{intersection}(w_4, x_4, y_4, w_6)$	$(Step7b4)$
$\vee$	$\mathsf{inside}(z_3, w_4, y_4, w_6)$	$(d_6(Z) : Step7b\ with\ w_6\ as\ z_4,$ $d_6(Y) : Step8b)$
$\vee$	$\mathsf{inside}(z_3, w_4, y_4, x_6)$	$(d_6(Z) : Step7b\ with\ x_6\ as\ z_4,$ $d_6(Y) : Step8b)$

Step7b4:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4)$	$(e_4 = y_4 w_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_4) \wedge \mathsf{E}(y_4, z_4) \wedge \mathsf{inside}(z_3, w_4, y_4, z_4)$	$(e_5 = y_4 z_4)$
$\wedge\neg$	$\mathsf{intersection}(w_4, x_4, y_4, z_4)$	$(\leftarrow Step7a2)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, w_6, x_6)$	$(d_{6a} = w_6 x_6)$
$\wedge$	$\mathsf{deleting}(y_4, z_4, x_6, w_6)$	$(d_{6b} = x_6 w_6)$
$\wedge$	$\mathsf{intersection}(w_4, x_4, y_4, w_6)$	
$\rightarrow$	$\mathsf{deleting}(y_4, z_3, w_6, x_4)$	$(d'_4(Z) = w_6 x_4 (Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, w_6, x_6)$	$(d'_4(Z) = w_6 x_6 (Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_6, x_4)$	$(d'_4(Z) = x_6 x_4 (Step6))$
$\vee$	$\mathsf{deleting}(y_4, z_3, x_6, w_6)$	$(d'_4(Z) = x_6 w_6 (Step6))$

Step7c1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1)$	$(e_1 = y_2 z_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3)$	$(e_2 = y_2 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4)$	$(e_3 = y_4 z_3)$
$\wedge\neg$	$\mathsf{intersection}(y_2, z_1, y_4, z_3)$	$(\leftarrow Step2c2)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(y_2, z_1, y_4, w_4)$	
$\rightarrow$	$\bot$	

Step7c2:

$$
\begin{array}{lr}
\quad \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge \ \mathsf{deleting}(y_4, z_3, w_4, x_4) & (d_4(Z) = w_4 x_4) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_4, w_4) \wedge \mathsf{E}(y_4, w_4) & (e_4 = y_4 w_4) \\
\wedge \neg \mathsf{intersection}(y_2, z_1, y_4, w_4) & (\leftarrow Step7c1) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, y_4, w_4) & (\leftarrow Step6a2) \\
\wedge \neg \mathsf{inside}(y_2, y_4, z_3, w_4) & (\leftarrow Step6c) \\
\wedge \ \mathsf{inside}(z_3, w_4, y_4, z_4) & \\
\rightarrow &
\end{array}
$$

Step7c3:

$$
\begin{array}{lr}
\quad \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge \ \mathsf{deleting}(y_4, z_3, w_4, x_4) & (d_4(Z) = w_4 x_4) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_4, w_4) \wedge \mathsf{E}(y_4, w_4) & (e_4 = y_4 w_4) \\
\wedge \neg \mathsf{intersection}(y_2, z_1, y_4, w_4) & (\leftarrow Step7c1) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, y_4, w_4) & (\leftarrow Step6a2) \\
\wedge \neg \mathsf{inside}(y_2, y_4, z_3, w_4) & (\leftarrow Step6c) \\
\wedge \ \mathsf{inside}(w_4, z_3, y_4, z_4) & \\
\wedge \ \mathsf{intersection}(y_2, z_1, y_4, z_4) & \\
\rightarrow \ \bot &
\end{array}
$$

Step7c4:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) & (d_4(Z) = w_4 x_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_4) \wedge \mathsf{E}(y_4, w_4) & (e_4 = y_4 w_4) \\
\wedge\neg & \mathsf{intersection}(y_2, z_1, y_4, w_4) & (\leftarrow Step7c1) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, y_4, w_4) & (\leftarrow Step6a2) \\
\wedge\neg & \mathsf{inside}(y_2, y_4, z_3, w_4) & (\leftarrow Step6c) \\
\wedge & \mathsf{inside}(w_4, z_3, y_4, z_4) & \\
\wedge & \mathsf{intersection}(y_2, z_3, y_4, z_4) & \\
\rightarrow & \bot &
\end{array}
$$

Step7d1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4 z_5) \\
\wedge\neg & \mathsf{intersection}(y_2, z_1, y_4, z_5) & (\leftarrow Step6a2 \ for \ k-1) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, y_4, z_5) & (\leftarrow Step6a3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_5, z_3, y_4, y_6) & (e_5 = y_6 z_5) \\
\wedge\neg & \mathsf{intersection}(y_2, z_1, y_6, z_5) & (\leftarrow Step8c2 \ for \ k-1) \\
\wedge\neg & \mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step8c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_6, z_5, w_6, x_6) & (d_6(Z) = w_6 x_6) \\
\wedge\neg & \mathsf{F}(w_6, x_6) & (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(y_2, z_1, y_6, w_6) & \\
\rightarrow & \bot &
\end{array}
$$

Step7d2:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_5, z_3, y_4, y_6) & (e_5 = y_6 z_5) \\
\wedge & \mathsf{deleting}(y_6, z_5, w_6, x_6) & (d_6(Z) = w_6 x_6) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_6) & (e_6 = y_6 w_6) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_6, w_6) & \\
\wedge & \neg\mathsf{intersection}(y_4, z_3, y_6, w_6) & (\leftarrow Step7c - f \ for \ j+1) \\
\rightarrow & \bot &
\end{array}
$$

Step7d3:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_5) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6) & (e_5 = y_6 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_5) & (\leftarrow Step8c2 \ for \ k-1) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step8c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_6, z_5, w_6, x_6) & (d_6(Z) = w_6 x_6) \\
\wedge & \neg\mathsf{F}(w_6, x_6) & (\leftarrow Step6b) \\
\wedge & \mathsf{inside}(z_5, w_6, y_6, z_1) & \\
\rightarrow & \bot &
\end{array}
$$

Step7d4:

$\mathsf{E}(u_1, v_1)$ $(e_0 = u_1 v_1)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1)$ $(e_1 = y_2 z_1)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$ $(e_2 = y_2 z_3)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_2, z_5)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6)$ $(e_5 = y_6 z_5)$

$\wedge\neg\mathsf{intersection}(y_2, z_1, y_6, z_5)$ $(Step8c2\ for\ k-1)$

$\wedge\neg\mathsf{intersection}(y_2, z_3, y_6, z_5)$ $(Step8c3\ for\ k-1)$

$\wedge$ $\mathsf{deleting}(y_6, z_5, w_6, x_6)$ $(d_6(Z) = w_6 x_6)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_6, z_6)$ $(e_6 = y_6 z_6)$

$\wedge\neg\mathsf{intersection}(y_2, z_1, y_6, w_6)$ $(\leftarrow Step3d1)$

$\wedge\neg\mathsf{intersection}(y_2, z_3, y_6, w_6)$ $(\leftarrow Step3d2)$

$\wedge\neg\mathsf{inside}(z_5, w_6, y_6, z_1)$ $(\leftarrow Step3d3)$

$\wedge$ $\mathsf{inside}(z_5, w_6, y_6, z_6)$

$\wedge$ $\mathsf{intersection}(y_2, z_1, y_6, z_6)$

$\rightarrow$ $\perp$

Step7e1:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) && (e_1 = y_2 z_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_5) && \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6) && (e_5 = y_6 z_5) \\
\wedge & \neg \mathsf{intersection}(y_2, z_1, y_6, z_5) && (\leftarrow Step8c2\ for\ k-2) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_5) && (\leftarrow Step8c3\ for\ k-2) \\
\wedge\ & \mathsf{deleting}(y_6, z_5, z_7, x_6) && (d_6(Z) = z_7 x_6) \\
\wedge & \neg \mathsf{F}(z_7, x_6) && (\leftarrow Step6b) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_6, z_7) && (e_6 = y_6 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_1, y_6, z_7) && (\leftarrow Step6a2\ for\ k-2) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_6, z_7) && (\leftarrow Step6a3\ for\ k-2) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_5, z_7, y_6, y_8) && (e_7 = y_8 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_1, y_8, z_7) && (\leftarrow Step8c2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_7) && (\leftarrow Step8c3\ for\ k-1) \\
\wedge\ & \mathsf{deleting}(y_8, z_7, w_8, x_8) && (d_8(Z) = w_8 x_8) \\
\wedge & \neg \mathsf{F}(w_8, x_8) && (\leftarrow Step6b) \\
\wedge\ & \mathsf{intersection}(y_2, z_1, y_8, w_8) && \\
\rightarrow\ & \bot &&
\end{aligned}
$$

Step7e2:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_5) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6) & (e_5 = y_6 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_5) & (\leftarrow Step8c2 \ for \ k-2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step8c3 \ for \ k-2) \\
\wedge & \mathsf{deleting}(y_6, z_5, w_6, x_6) & (d_6(Z) = w_6 x_6) \\
\wedge & \neg\mathsf{F}(w_6, x_6) & (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, w_6) & (e_6 = y_6 w_6) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(w_6, z_5, y_6, z_7) & (e_7 = y_6 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_7) & (\leftarrow Step6a2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step6a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_5, z_7, y_6, y_8) & (e_8 = y_8 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_8, z_7) & (\leftarrow Step8c2 \ for \ k-1) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step8c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_8, z_7, w_9, x_9) & (d_9(Z) = w_9 x_9) \\
\wedge & \neg\mathsf{F}(w_9, x_9) & (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(y_2, z_1, y_8, w_9) & \\
\rightarrow & \bot &
\end{array}
$$

Step7e3:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_5, z_3, y_4, y_6) & (e_5 = y_6 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_7) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_5, z_3, y_4, y_8) & (e_7 = y_8 z_7) \\
\wedge\neg & \mathsf{intersection}(y_4, z_3, y_8, z_7) & (\leftarrow Step8c2 \ for \ k-1) \\
\wedge\neg & \mathsf{intersection}(y_4, z_5, y_8, z_7) & (\leftarrow Step8c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_8, z_7, w_8, x_8) & (d_8(Z) = w_8 x_8) \\
\wedge\neg & \mathsf{F}(w_8, x_8) & (\leftarrow Step6b) \\
\wedge & \mathsf{inside}(w_8, z_7, y_8, z_3) & \\
\rightarrow \vee & \bot &
\end{array}
$$

Step7e4:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_5) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6) & (e_5 = y_6 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_5) & (\leftarrow Step8c2 \ for \ k-2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step8c3 \ for \ k-2) \\
\wedge & \mathsf{deleting}(y_6, z_5, z_7, x_6) & (d_6(Z) = z_7 x_6) \\
\wedge & \neg\mathsf{F}(z_7, x_6) & (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) & (e_6 = y_6 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_7) & (\leftarrow Step6a2 \ for \ k-2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step6a3 \ for \ k-2) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_5, z_7, y_6, y_8) & (e_7 = y_8 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_8, z_7) & (\leftarrow Step8c2 \ for \ k-1) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step8c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_8, z_7, w_8, x_8) & (d_8(Z) = w_8 x_8) \\
\wedge & \neg\mathsf{F}(w_8, x_8) & (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, w_8) & (e_8 = y_8 w_8) \\
\wedge & \mathsf{inside}(w_8, z_7, y_8, z_8) & \\
\wedge & \mathsf{intersection}(y_2, z_1, y_8, z_8) & \\
\rightarrow & \bot &
\end{array}
$$

Step7e5:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_5) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6) & (e_5 = y_6 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_5) & (\leftarrow Step8c2 \ for \ k-2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step8c3 \ for \ k-2) \\
\wedge & \mathsf{deleting}(y_6, z_5, w_6, x_6) & (d_6(Z) = w_6 x_6) \\
\wedge & \neg\mathsf{F}(w_6, x_6) & (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, w_6) & (e_6 = y_6 w_6) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(z_5, w_6, y_6, z_7) & (e_7 = y_6 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_7) & (\leftarrow Step6a2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step6a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_5, z_7, y_6, y_8) & (e_8 = y_8 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_8, z_7) & (\leftarrow Step8c2 \ for \ k-1) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step8c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_8, z_7, w_9, x_9) & (d_9(Z) = w_9 x_9) \\
\wedge & \neg\mathsf{F}(w_9, x_9) & (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, w_9) & (e_9 = y_8 w_9) \\
\wedge & \mathsf{inside}(w_9, z_7, y_8, z_8) & \\
\wedge & \mathsf{intersection}(y_2, z_1, y_8, z_8) & \\
\rightarrow & \bot &
\end{array}
$$

Step7f1:

$$
\begin{aligned}
& \mathsf{E}(u_1, v_1) && (e_0 = u_1 v_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) && (e_1 = y_2 z_1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) && (e_2 = y_2 z_3) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_2, z_7) && \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_1, z_3, y_2, y_8) && (e_7 = y_8 z_7) \\
\wedge & \neg \mathsf{intersection}(y_2, z_1, y_8, z_7) && (\leftarrow Step8c2\ for\ k-2) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_7) && (\leftarrow Step8c3\ for\ k-2) \\
\wedge\ & \mathsf{deleting}(y_8, z_7, z_9, x_8) && (d_8(Z) = z_9 x_8) \\
\wedge & \neg \mathsf{F}(z_9, x_8) && (Step6b\ for\ k-1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_8, z_9) && (e_8 = y_8 z_9) \\
\wedge & \neg \mathsf{intersection}(y_2, z_1, y_8, z_9) && (\leftarrow Step6a2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_8, z_9) && (\leftarrow Step6a3\ for\ k-1) \\
\wedge\ & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) \wedge \mathsf{inside}(z_9, z_7, y_8, y_{10}) && (e_9 = y_{10} z_9) \\
\wedge & \neg \mathsf{intersection}(y_2, z_1, y_{10}, z_9) && (\leftarrow Step8c2\ for\ k-1) \\
\wedge & \neg \mathsf{intersection}(y_2, z_3, y_{10}, z_9) && (\leftarrow Step8c3\ for\ k-1) \\
\wedge\ & \mathsf{deleting}(y_{10}, z_9, w_{10}, x_{10}) && (d_{10}(Z) = w_{10} x_{10}) \\
\wedge & \neg \mathsf{F}(w_{10}, x_{10}) && (\leftarrow Step6b) \\
\wedge\ & \mathsf{intersection}(y_2, z_1, y_{10}, w_{10}) && \\
\rightarrow\ & \bot &&
\end{aligned}
$$

Step7f2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) \qquad (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_1, z_3, y_2, y_8) \qquad (e_7 = y_8 z_7) \\
\wedge \neg & \mathsf{intersection}(y_2, z_1, y_8, z_7) \qquad (\leftarrow Step8c2 \ for \ k-2) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_8, z_7) \qquad (\leftarrow Step8c3 \ for \ k-2) \\
\wedge & \mathsf{deleting}(y_8, z_7, w_8, x_8) \qquad (d_8(Z) = w_8 x_8) \\
\wedge \neg & \mathsf{F}(w_8, x_8) \qquad (\leftarrow Step6b \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, w_9) \qquad (e_8 = y_8 w_9) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(w_8, z_7, y_8, z_9) \qquad (e_9 = y_8 z_9) \\
\wedge \neg & \mathsf{intersection}(y_2, z_1, y_8, z_9) \qquad (\leftarrow Step6a2) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_8, z_9) \qquad (\leftarrow Step6a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) \wedge \mathsf{inside}(z_9, z_7, y_8, y_{10}) \qquad (e_{10} = y_{10} z_9) \\
\wedge \neg & \mathsf{intersection}(y_2, z_1, y_{10}, z_9) \qquad (\leftarrow Step8c2 \ for \ k-1) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_{10}, z_9) \qquad (\leftarrow Step8c3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_{10}, z_9, w_{11}, x_{11}) \qquad (d_{11}(Z) = w_{11} x_{11}) \\
\wedge \neg & \mathsf{F}(w_{11}, x_{11}) \qquad (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(y_2, z_1, y_{10}, w_{11}) \\
\rightarrow & \ \bot
\end{array}
$$

Step8a1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, z_5, x_4) \qquad (d_4(Z) = z_5 x_4) \\
\wedge \neg & \mathsf{F}(z_5, x_4) \qquad (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \qquad (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) \qquad (d_5(Y) = w_5 x_5) \\
\rightarrow & \mathsf{deleting}(y_4, z_5, x_5, w_5) \qquad (d_{5b} = x_5 w_5 (Step8a3)) \\
\vee & \mathsf{inside}(z_3, z_5, y_4, w_5) \qquad (Step8b) \\
\vee & \mathsf{inside}(z_5, z_3, y_4, w_5) \qquad (Step8b)
\end{array}
$$

Step8a2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(z_3, w_4, y_4, z_5)$	$(e_5 = y_4 z_5)$
$\wedge\neg$	$\mathsf{intersection}(w_4, x_4, y_4, z_5)$	$(\leftarrow Step7a2/7b2/7b4)$
$\wedge$	$\mathsf{deleting}(y_4, z_5, w_6, x_6)$	$(d_6(Y) = w_6 x_6)$
$\rightarrow$	$\mathsf{deleting}(y_4, z_5, x_6, w_6)$	$(d_{6b} = x_6 w_6(Step8a3))$
$\vee$	$\mathsf{inside}(z_3, z_5, y_4, w_5)$	$(Step8b)$

Step8a3:

	$\mathsf{E}(u_1, v_1) \wedge \neg\mathsf{F}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{E}(z_3, z_5)$	$(e_4 = y_4 z_5)$
$\wedge$	$\mathsf{deleting}(y_4, z_5, w_5, x_5)$	$(d_{5a}(Y) = w_6 x_6)$
$\wedge$	$\mathsf{deleting}(y_4, z_5, x_5, w_5)$	$(d_{5b}(Y) = x_5 w_5)$
$\rightarrow$	$\mathsf{inside}(z_3, z_5, y_4, w_5) \vee \mathsf{inside}(z_5, z_3, y_4, w_5)$	$(Step8b)$
$\vee$	$\mathsf{inside}(z_3, z_5, y_4, x_5) \vee \mathsf{inside}(z_5, z_3, y_4, x_5)$	$(Step8b)$

Step8a4:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, z_5, x_4)$	$(d_4(Z) = y_4 z_5)$
$\wedge\neg$	$\mathsf{F}(z_5, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_5)$	$(e_4 = y_4 z_5)$
$\wedge$	$\mathsf{deleting}(y_4, z_5, w_5, x_5)$	$(d_{5a}(Y) = w_5 x_5)$
$\wedge$	$\mathsf{deleting}(y_4, z_5, x_5, w_5)$	$(d_{5b}(Y) = x_5 w_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_5, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, w_5)$	$(d_5 = d_{5a}, e_5 = w_5 z_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, x_5, z_5)$	$(d_5 = d_{5b}, e_5 = x_5 z_5)$
$\wedge$	$\mathsf{deleting}(w_5, z_5, w_6, x_6)$	$(d_5 = d_{5a}, d_6(Y) = w_6 x_6)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_6, z_5)$	$(d_5 = d_{5a}, e_6 = w_6 z_5)$
$\wedge$	$\mathsf{deleting}(w_6, z_5, w_5, x_5)$	$(d_5 = d_{5a}, d_7(Y) = w_5 x_5)$
$\rightarrow$	$\mathsf{inside}(z_3, z_5, y_4, x_5)$	$(d_5 = d_{5b}(Step8b))$

Step8a5:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) \qquad (d_4(Z) = w_4 x_4) \\
\wedge & \neg\mathsf{F}(w_4, x_4) \qquad (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(z_3, w_4, y_4, z_5) \qquad (e_5 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_6, x_6) \qquad (d_{6a}(Y) = w_6 x_6) \\
\wedge & \mathsf{deleting}(y_4, z_5, x_6, w_6) \qquad (d_{6b}(Y) = x_6 w_6) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_6, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, w_6) \qquad (d_6 = d_{6a}, e_5 = w_6 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, x_6, z_5) \qquad (d_6 = d_{6b}, e_5 = w_6 z_5) \\
\wedge & \mathsf{deleting}(w_6, z_5, w_7, x_7) \qquad (d_6 = d_{6a}, d_7(Y) = w_7 x_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_7, z_5) \qquad (d_6 = d_{6a}, e_7 = w_7 z_5) \\
\wedge & \mathsf{deleting}(w_7, z_5, w_6, x_6) \qquad (d_6 = 6a, d_8(Y) = w_6 x_6) \\
\rightarrow & \mathsf{inside}(z_3, z_5, y_4, x_6) \qquad (d_6 = d_{6b}(Step8b))
\end{array}
$$

Step8a6:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \wedge \neg\mathsf{F}(u_1, v_1) \qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{E}(z_3, z_5) \qquad (e_4 = y_4 z_5) \\
\wedge & \mathsf{deleting}(y_4, z_5, w_5, x_5) \qquad (d_5(Y) = w_5 x_5) \\
\wedge & \mathsf{intersection}(y_4, z_3, w_5, z_5) \\
\rightarrow & \bot
\end{array}
$$

Step8b1:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, z_5, x_4)$	$(d_4(Z) = z_5 x_4)$
$\wedge\neg$	$\mathsf{F}(z_5, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_5)$	$(e_4 = y_4 z_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_5, z_5) \wedge \mathsf{E}(y_5, z_5) \wedge \mathsf{inside}(z_5, z_3, y_4, y_5)$	$(e_5 = y_5 z_5)$
$\wedge$	$\mathsf{deleting}(y_5, z_5, w_6, x_6)$	$(d_6 = w_6 x_6)$
$\rightarrow$	$\mathsf{deleting}(y_5, z_5, x_6, w_6)$	$(d_{6b} = x_6 w_6 (Step8b3))$
$\vee$	$\mathsf{inside}(z_3, z_5, y_4, w_6)$	$(d_6(Y) : Step8b\ with\ w_6\ as\ y_5,$
		$d_6(Z) : Step6 for\ k+1)$
$\vee$	$\mathsf{intersection}(z_3, z_5, y_5, w_6)$	$(d_6(Z) : Step7 for\ k+1)$

Step8b2:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, w_4, x_4)$	$(d_4(Z) = w_4 x_4)$
$\wedge\neg$	$\mathsf{F}(w_4, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, w_4)$	$(e_4 = y_4 w_4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(z_3, w_4, y_4, z_5)$	$(e_5 = y_4 z_5)$
$\wedge\neg$	$\mathsf{intersection}(w_4, x_4, y_4, z_5)$	$(\leftarrow Step7a2/7b2/7b4)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_5, z_5) \wedge \mathsf{E}(y_5, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, y_5)$	$(e_6 = y_5 z_5)$
$\wedge$	$\mathsf{deleting}(y_5, z_5, w_7, x_7)$	$(d_7 = w_7 x_7)$
$\rightarrow$	$\mathsf{deleting}(y_5, z_5, x_7, w_7)$	$(d_{7b} = x_7 w_7 (Step8b3))$
$\vee$	$\mathsf{inside}(z_3, z_5, y_4, w_7)$	$(d_6(Y) : Step8b\ with\ w_7\ as\ y_5,$
		$d_6(Z) : Step6 for\ k+1)$
$\vee$	$\mathsf{intersection}(z_3, z_5, y_5, w_7)$	$(Step7 for\ k+1)$

Step8b3:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{E}(z_3, z_5)$	$(e_4 = y_4 z_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_5, z_5) \wedge \mathsf{E}(y_5, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, y_5)$	$(e_5 = y_5 z_5)$
$\wedge$	$\mathsf{deleting}(y_5, z_5, w_6, x_6)$	$(d_{6a} = w_6 x_6)$
$\wedge$	$\mathsf{deleting}(y_5, z_5, x_6, w_6)$	$(d_{6b} = x_6 w_6)$
$\rightarrow$	$\mathsf{inside}(z_3, z_5, y_4, w_6)$	$(d_6(Y) : Step8b\ with\ w_6\ as\ y_5,$ $d_6(Z) : Step6\ for\ k+1)$
$\vee$	$\mathsf{inside}(z_3, z_5, y_4, x_6)$	$(d_6(Y) : Step8b\ with\ x_6\ as\ y_5,$ $d_6(Z) : Step6\ for\ k+1)$

Step8b4:

	$\mathsf{E}(u_1, v_1)$	$(e_0 = u_1 v_1)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3)$	$(e_3 = y_4 z_3)$
$\wedge$	$\mathsf{deleting}(y_4, z_3, z_5, x_4)$	$(d_4(Z) = z_5 x_4)$
$\wedge$	$\neg\mathsf{F}(z_5, x_4)$	$(\leftarrow Step6b)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_4, z_5)$	$(e_4 = y_4 z_5)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, y_5, z_5) \wedge \mathsf{E}(y_5, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, y_5)$	$(e_5 = y_5 z_5)$
$\wedge$	$\mathsf{deleting}(y_5, z_5, w_6, x_6)$	$(d_6(Y) = w_6 x_6)$
$\wedge$	$\mathsf{intersection}(u_1, v_1, w_6, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, w_6)$	$(e_6 = w_6 z_5)$
$\wedge$	$\mathsf{intersection}(y_4, z_3, w_6, z_5)$	
$\rightarrow$	$\perp$	

Step8b5:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{deleting}(y_4, z_3, w_4, x_4) & (d_4(Z) = w_4 x_4) \\
\wedge \neg & \mathsf{F}(w_4, x_4) & (\leftarrow Step6b) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, w_4) & (e_4 = y_4 w_4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(z_3, w_4, y_4, z_5) & (e_5 = y_4 z_5) \\
\wedge \neg & \mathsf{intersection}(w_4, x_4, y_4, z_5) & (\leftarrow Step7b2/7b4) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_5, z_5) \wedge \mathsf{E}(y_5, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, y_5) & (e_6 = y_5 z_5) \\
\wedge & \mathsf{deleting}(y_5, z_5, w_7, x_7) & (d_7(Y) = w_7 x_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_7, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, w_7) & (e_7 = w_7 z_5) \\
\wedge & \mathsf{intersection}(y_4, z_3, w_7, z_5) & \\
\rightarrow & \bot &
\end{array}
$$

Step8c1:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4 z_5) \\
\wedge \neg & \mathsf{intersection}(y_2, z_1, y_4, z_5) & (\leftarrow Step6a2) \\
\wedge \neg & \mathsf{intersection}(y_2, z_3, y_4, z_5) & (\leftarrow Step6a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_5, z_3, y_4, y_6) & (e_5 = y_6 z_5) \\
\rightarrow & \mathsf{inside}(z_1, z_3, y_2, y_6) &
\end{array}
$$

Step8c2:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_4, z_5) & (\leftarrow Step6a2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_4, z_5) & (\leftarrow Step6a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_5, z_3, y_4, y_6) & (e_5 = y_6 z_5) \\
\wedge & \mathsf{intersection}(y_2, z_1, y_6, z_5) & \\
\rightarrow & \bot &
\end{array}
$$

Step8c3:

$$
\begin{array}{ll}
& \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{inside}(z_1, z_3, y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) & (e_4 = y_4 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_4, z_5) & (\leftarrow Step6a2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_4, z_5) & (\leftarrow Step6a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_5, z_3, y_4, y_6) & (e_5 = y_6 z_5) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_6, z_5) & \\
\rightarrow & \bot &
\end{array}
$$

Step8d1:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_5) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6) & (e_5 = y_6 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_5) & (\leftarrow Step8c2/e2 \; for \; k-1) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step8c3/e3 \; for \; k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) & (e_6 = y_6 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_7) & (\leftarrow Step6a2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step6a3) \\
\wedge & \neg\mathsf{inside}(z_5, z_7, y_6, z_1) & (\leftarrow Step7c3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_5, z_7, y_6, y_8) & (e_7 = y_8 z_7) \\
\rightarrow & \mathsf{inside}(z_1, z_3, y_2, y_8) &
\end{array}
$$

Step8d2:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_5) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6) & (e_5 = y_6 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_5) & (\leftarrow Step8c2/e2 \; for \; k-1) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step8c3/e3 \; for \; k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) & (e_6 = y_6 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_7) & (\leftarrow Step6a2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step6a3) \\
\wedge & \neg\mathsf{inside}(z_5, z_7, y_6, z_1) & (\leftarrow Step7c3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_5, z_7, y_6, y_8) & (e_7 = y_8 z_7) \\
\wedge & \mathsf{intersection}(y_2, z_1, y_8, z_7) & \\
\rightarrow & \bot &
\end{array}
$$

Step8d3:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_5) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_1, z_3, y_2, y_6) & (e_5 = y_6 z_5) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_5) & (\leftarrow Step8c2/e2 \; for \; k-1) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_5) & (\leftarrow Step8c3/e3 \; for \; k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) & (e_6 = y_6 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_6, z_7) & (\leftarrow Step6a2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_6, z_7) & (\leftarrow Step6a3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_5, z_7, y_6, y_8) & (e_7 = y_8 z_7) \\
\wedge & \mathsf{intersection}(y_2, z_3, y_8, z_7) & \\
\rightarrow & \bot &
\end{array}
$$

Step8e1:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_7) & \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_1, z_3, y_2, y_8) & (e_7 = y_8 z_7) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_8, z_7) & (\leftarrow Step8c2/e2 \; for \; k-1) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step8c3/e3 \; for \; k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) & (e_8 = y_8 z_9) \\
\wedge & \neg\mathsf{intersection}(y_2, z_1, y_8, z_9) & (\leftarrow Step6a2) \\
\wedge & \neg\mathsf{intersection}(y_2, z_3, y_8, z_9) & (\leftarrow Step6a3) \\
\wedge & \neg\mathsf{inside}(z_9, z_7, y_8, z_3) & (\leftarrow Step7d3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) \wedge \mathsf{inside}(z_9, z_7, y_8, y_{10}) & (e_9 = y_{10} z_9) \\
\rightarrow & \mathsf{inside}(z_1, z_3, y_2, y_{10}) &
\end{array}
$$

Step8e2:

$$
\begin{array}{ll}
\quad \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_7) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_1, z_3, y_2, y_8) & (e_7 = y_8 z_7) \\
\wedge \neg \mathsf{intersection}(y_2, z_1, y_8, z_7) & (\leftarrow Step8c2/e2 \ for \ k-1) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step8c3/e3 \ for \ k-1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) & (e_8 = y_8 z_9) \\
\wedge \neg \mathsf{intersection}(y_2, z_1, y_8, z_9) & (\leftarrow Step6a2) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, y_8, z_9) & (\leftarrow Step6a3) \\
\wedge \neg \mathsf{inside}(z_9, z_7, y_8, z_3) & (\leftarrow Step7d3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) \wedge \mathsf{inside}(z_9, z_7, y_8, y_{10}) & (e_9 = y_{10} z_9) \\
\wedge \ \mathsf{intersection}(y_2, z_1, y_{10}, z_9) & \\
\rightarrow \ \bot &
\end{array}
$$

Step8e3:

$$
\begin{array}{ll}
\quad \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_1) \wedge \mathsf{E}(y_2, z_1) & (e_1 = y_2 z_1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \wedge \mathsf{E}(z_1, z_3) & (e_2 = y_2 z_3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_2, z_7) & \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_1, z_3, y_2, y_8) & (e_7 = y_8 z_7) \\
\wedge \neg \mathsf{intersection}(y_2, z_1, y_8, z_7) & (\leftarrow Step8c2/e2 \ for \ k-1) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, y_8, z_7) & (\leftarrow Step8c3/e3 \ for \ k-1) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) & (e_8 = y_8 z_9) \\
\wedge \neg \mathsf{intersection}(y_2, z_1, y_8, z_9) & (\leftarrow Step6a2) \\
\wedge \neg \mathsf{intersection}(y_2, z_3, y_8, z_9) & (\leftarrow Step6a3) \\
\wedge \neg \mathsf{inside}(z_9, z_7, y_8, z_3) & (\leftarrow Step7d3) \\
\wedge \ \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}, z_9) \wedge \mathsf{inside}(z_9, z_7, y_8, y_{10}) & (e_9 = y_{10} z_9) \\
\wedge \ \mathsf{intersection}(y_2, z_3, y_{10}, z_9) & \\
\rightarrow \ \bot &
\end{array}
$$

Step9a:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_2, y_4, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, y_6) & (e_5 = y_6 z_5) \\
\wedge & \mathsf{deleting}(y_6, z_5, w_6, x_6) & (d_6(Z) = w_6 x_6) \\
\rightarrow & \mathsf{F}(w_6, x_6) & (\rightarrow \mathit{Step6b}) \\
\vee & \mathsf{inside}(y_2, y_4, z_3, w_6) &
\end{array}
$$

Step9b:

$$
\begin{array}{lll}
 & \mathsf{E}(u_1, v_1) & (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) & (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) & (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_2, y_4, z_3, z_5) & (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_5) \wedge \mathsf{E}(y_6, z_5) \wedge \mathsf{inside}(z_3, z_5, y_4, y_6) & (e_5 = y_6 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_6, z_7) \wedge \mathsf{E}(y_6, z_7) \wedge \mathsf{inside}(y_2, y_4, z_3, z_7) & (e_6 = y_6 z_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_7, z_5, y_6, y_8) & (e_7 = y_8 z_7) \\
\wedge & \mathsf{deleting}(y_8, z_7, w_8, x_8) & (d_8(Z) = w_8 x_8) \\
\rightarrow & \mathsf{F}(w_8, x_8) & (\rightarrow \mathit{Step6b}) \\
\vee & \mathsf{inside}(y_2, y_4, z_3, w_8) &
\end{array}
$$

Step9c1:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_2, y_4, z_3, z_5) \qquad\qquad (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_7) \wedge \mathsf{inside}(y_2, y_4, z_3, z_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_3, z_5, y_4, y_8) \qquad\qquad (e_7 = y_8 z_7) \\
\wedge\neg & \mathsf{intersection}(y_4, z_5, y_8, z_7) \qquad\qquad (\leftarrow Step8c3/e3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_8, z_7, z_9, x_8) \qquad\qquad (d_8(Z) = z_9 x_8) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{inside}(y_2, y_4, z_3, z_9) \qquad\qquad (e_8 = y_8 z_9) \\
\wedge\neg & \mathsf{intersection}(y_4, z_5, y_8, z_9) \qquad\qquad (\leftarrow Step8c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(z_9, y_{10}) \wedge \mathsf{inside}(z_7, z_9, y_8, y_{10}) \qquad\qquad (e_9 = y_{10} z_9) \\
\wedge & \mathsf{deleting}(z_9, y_{10}, w_{10}, x_{10}) \qquad\qquad (d_{10}(Z) = w_{10} x_{10}) \\
\rightarrow & \mathsf{F}(w_{10}, x_{10}) \qquad\qquad (\rightarrow Step6b) \\
\vee & \mathsf{inside}(y_2, y_4, z_3, w_{10})
\end{array}
$$

Step9c2:

$$
\begin{array}{ll}
 & \mathsf{E}(u_1, v_1) \qquad\qquad (e_0 = u_1 v_1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3) \qquad\qquad (e_2 = y_2 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4) \qquad\qquad (e_3 = y_4 z_3) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_2, y_4, z_3, z_5) \qquad\qquad (e_4 = y_4 z_5) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_4, z_7) \wedge \mathsf{inside}(y_2, y_4, z_3, z_7) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_7) \wedge \mathsf{E}(y_8, z_7) \wedge \mathsf{inside}(z_3, z_5, y_4, y_8) \qquad\qquad (e_7 = y_8 z_7) \\
\wedge\neg & \mathsf{intersection}(y_4, z_5, y_8, z_7) \qquad\qquad (\leftarrow Step8c3/e3 \ for \ k-1) \\
\wedge & \mathsf{deleting}(y_8, z_7, w_8, x_8) \qquad\qquad (d_8(Z) = w_8 x_8) \\
\wedge & \mathsf{intersection}(u_1, v_1, w_8, z_9) \wedge \mathsf{inside}(y_2, y_4, z_3, w_8) \qquad\qquad (e_8 = w_8 z_9) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_8, z_9) \wedge \mathsf{E}(y_8, z_9) \wedge \mathsf{inside}(z_7, w_8, y_8, z_9) \qquad\qquad (e_9 = y_8 z_9) \\
\wedge\neg & \mathsf{intersection}(y_4, z_5, y_8, z_9) \qquad\qquad (\leftarrow Step8c3/e3 \ for \ k-1) \\
\wedge & \mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(y_{10}.z_9) \wedge \mathsf{inside}(z_7, z_9, y_8, y_{10}) \qquad\qquad (e_{10} = y_{10} z_9) \\
\wedge & \mathsf{deleting}(y_{10}, z_9, w_{11}, x_{11}) \qquad\qquad (d_{11}(Z) = w_{11} x_{11}) \\
\rightarrow & \mathsf{F}(w_{11}, x_{11}) \qquad\qquad (\rightarrow Step6b) \\
\vee & \mathsf{inside}(y_2, y_4, z_3, w_{11})
\end{array}
$$

Step9d:

$\mathsf{E}(u_1, v_1)$ $(e_0 = u_1 v_1)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_2, z_3) \wedge \mathsf{E}(y_2, z_3)$ $(e_2 = y_2 z_3)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_4, z_3) \wedge \mathsf{E}(y_4, z_3) \wedge \mathsf{E}(y_2, y_4)$ $(e_3 = y_4 z_3)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_4, z_5) \wedge \mathsf{E}(y_4, z_5) \wedge \mathsf{inside}(y_2, y_4, z_3, z_5)$ $(e_4 = y_4 z_5)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_4, z_9) \wedge \mathsf{inside}(y_2, y_4, z_3, z_9)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_{10}, z_9) \wedge \mathsf{E}(z_9, y_{10}) \wedge \mathsf{inside}(z_3, z_5, y_4, y_{10})$ $(e_9 = y_{10} z_9)$

$\wedge\neg\mathsf{intersection}(y_4, z_5, y_{10}, z_9)$ $(\leftarrow Step8c3/e3\ for\ k-1)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_{10}, z_{11}) \wedge \mathsf{E}(z_{11}, y_{10}) \wedge \mathsf{inside}(y_2, y_4, z_3, z_{11})$ $(e_{10} = y_{10} z_{11})$

$\wedge\neg\mathsf{intersection}(y_4, z_5, y_{10}, z_{11})$ $(\leftarrow Step8c3/e3\ for\ k-1)$

$\wedge$ $\mathsf{intersection}(u_1, v_1, y_{12}, z_{11}) \wedge \mathsf{E}(y_{12}, z_{11}) \wedge \mathsf{inside}(z_{11}, z_9, y_{10}, y_{12})$ $(e_{11} = y_{12} z_{11})$

$\wedge$ $\mathsf{deleting}(y_{12}, z_{11}, w_{12}, x_{12})$ $(d_{12}(Z) = w_{12} x_{12})$

$\rightarrow$ $\mathsf{F}(w_{12}, x_{12})$ $(\rightarrow Step6b)$

$\vee$ $\mathsf{inside}(y_2, y_4, z_3, w_{12})$

9.4.6 Proof tasks of part4

Step1:

$\mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge \mathsf{F}(w_1, x_1)$

$\wedge$ $\mathsf{deleting}(v_1, w_1, w_2, x_2) \wedge \mathsf{F}(w_2, x_2)$

$\rightarrow$ $\mathsf{inside}(u_1, v_1, w_1, w_2) \vee \mathsf{inside}(v_1, u_1, w_1, w_2)$

$\vee$ $\mathsf{intersection}(u_1, v_1, w_2, x_2)$

$\vee$ $\mathsf{deleting}(v_1, w_1, x_2, w_2)$

Step2:

$\mathsf{deleting}(u_1, v_1, w_1, x_1) \wedge \mathsf{F}(w_1, x_1)$

$\wedge$ $\mathsf{deleting}(v_1, w_1, w_2, x_2) \wedge \mathsf{deleting}(v_1, w_1, x_2, w_2)$

$\rightarrow$ $\mathsf{inside}(u_1, v_1, w_1, w_2) \vee \mathsf{inside}(v_1, u_1, w_1, w_2)$

$\vee$ $\mathsf{inside}(u_1, v_1, w_1, x_2) \vee \mathsf{inside}(v_1, u_1, w_1, x_2)$

$\vee$ $\mathsf{intersection}(u_1, v_1, w_2, x_2)$

9.4.7 Proof tasks of part5

Step1:

$$
\begin{aligned}
& \mathsf{left}(u_1, v_1, w_1) \wedge \mathsf{left}(v_1, u_1, x_1) \\
\wedge\ & \mathsf{left}(w_1, x_1, v_1) \wedge \mathsf{left}(x_1, w_1, u_1) \\
\wedge\ & \mathsf{E}(u_1, v_1) \wedge \mathsf{F}(w_1, x_1) \\
\wedge\ & c_1 \neq \mathsf{nil} \wedge c_2 \neq \mathsf{nil} \wedge c_2 = \mathsf{next}(c_1) \\
\wedge\ & \mathsf{intersection}(c_1, c_2, a, b) \wedge F(a, b) \\
\rightarrow\ & \bot
\end{aligned}
$$

Step2:

$$
\begin{aligned}
& \mathsf{left}(u_1, v_1, w_1) \wedge \mathsf{left}(v_1, u_1, x_1) \\
\wedge\ & \mathsf{left}(w_1, x_1, v_1) \wedge \mathsf{left}(x_1, w_1, u_1) \\
\wedge\ & \mathsf{F}(u_1, v_1) \wedge \mathsf{E}(w_1, x_1) \\
\wedge\ & c_1 \neq \mathsf{nil} \wedge c_2 \neq \mathsf{nil} \wedge c_2 = \mathsf{next}(c_1) \\
\wedge\ & \mathsf{intersection}(c_1, c_2, a, b) \wedge F(a, b) \\
\rightarrow\ & \bot
\end{aligned}
$$