

Christian Koch

Technische Hochschule Nürnberg

Falsifizierung des Laplaceschen Dämons
mit Hilfe eines Computerspiels

Arbeitspapier

Nürnberg, 19.10.2018

Zusammenfassung

Diese Arbeit falsifiziert den Laplaceschen Dämon mit Hilfe eines Computerspiels. Das Spiel ist in der Programmiersprache C++ implementiert und basiert auf deterministischen Regeln. Es führt auf dem verarbeitenden Rechner zu einem probabilistischen Ergebnis. Geht man davon aus, dass der Laplacesche Dämon seinen schnellsten Rechner zur Ausführung des Spiels nutzt, oder es selbst ausführt, kann er die Ergebnisse nicht zu jedem Zeitpunkt vorhersagen. Da der Laplacesche Dämon dadurch definiert ist, dass er die Zukunft vollständig vorhersehen kann, ist er durch das Spiel widerlegt.

1. Einführung

Der „Laplacesche Dämon“ basiert auf einem Gedankenexperiment des französischen Mathematikers Pierre-Simon Laplace.¹ Konkret geht der Begriff auf folgendes Zitat des Mathematikers aus dem Jahre 1814 zurück (deutsche Übersetzung):²

„Wir müssen also den gegenwärtigen Zustand des Universums als Folge eines früheren Zustandes ansehen und als Ursache des Zustandes, der danach kommt. Eine Intelligenz, die in einem gegebenen Augenblick alle Kräfte kennt, mit denen die Welt begabt ist, und die gegenwärtige Lage der Gebilde, die sie zusammensetzen, und die überdies umfassend genug wäre, diese Kenntnisse der Analyse zu unterwerfen, würde in der gleichen Formel die Bewegungen der größten Himmelskörper und die des leichtesten Atoms einbegreifen. Nichts wäre für sie ungewiss, Zukunft und Vergangenheit lägen klar vor ihren Augen.“

Die oben genannte Intelligenz wird als Laplacescher Dämon bezeichnet. Geht man davon aus, dass die gesamte Welt auf deterministischen Regeln beruht, könnte der Dämon bei Kenntnis aller Regeln eine „Weltformel“ konstruieren. Nimmt man weiter an, ihm stehen alle Informationen zur Verfügung, könnte er die Formel nutzen, um die gesamte Vergangenheit zu erklären und die Zukunft vollständig vorherzusagen. Die Welt wäre deterministisch. Zufall gäbe es nicht.

Schaubildlich kann man sich den Laplaceschen Dämon als einen Supercomputer vorstellen. Er kennt die Position aller Atome des Universums.³ Zudem verfügt er über alle Regeln nach denen sich die Atome bewegen. Er wäre damit in der Lage die Historie des Universums zu rekonstruieren und alle zukünftigen Entwicklungen vorherzusagen. Sowohl aus mathematischer als auch aus physikalischer Sicht gibt es verschiedene Argumente gegen die Existenz des Laplaceschen Dämons.⁴ Dieser Aufsatz beschäftigt sich mit einer weiteren Möglichkeit ihn zu falsifizieren.

Die Arbeit geht von folgender Hypothese aus: Der Laplacesche Dämon ist falsifiziert, wenn es gelingt ein Spiel zu konstruieren, dass auf deterministischen Regeln

¹ Vgl. (Silver, 2012, S. 112-113)

² Zitiert nach: (Ferber, 2003, S. 178)

³ Das Wort „Atom“ wird stellvertretend für „kleinstes Teilchen“ verwendet.

⁴ Ein Beispiel ist die Heisenbergsche Unschärferelation. Siehe (Heisenberg, 1927)

basiert, jedoch zu einem probabilistischen Ergebnis führt. Ein solches Spiel wird im Folgenden „Laplacesches Spiel“ genannt.

2. Das Spiel

Folgendes Computerprogramm ist der Versuch eines Laplaceschen Spiels.

```
//Define a global variable x that stores the result of the program
long x = 0;

//Define functions a and b to set the value of x
void a();
void b();

//This function sets the value of x to 1 and calls b
void a(){
    x = 1;
    b();
}

//This function sets the value of x to 0 and calls a
void b(){
    x = 0;
    a();
}

//This main method executes the program and calls the function a
int main() {
    a();
    return 0;
}
```

Abbildung 1: Laplacesches Spiel

Inspiziert wurde das obige Programm durch den Gödelschen Unvollständigkeitssatz und das Lügner-Paradox.⁵ Die Software wurde mit Hilfe der Programmiersprache C++ implementiert.⁶

Das Spiel besteht aus einer Variablen x , deren Wert durch die beiden Funktionen a und b festgelegt wird. Die Funktion a setzt den Wert von x auf 1. Die Funktion b setzt den Wert von x auf 0. Die Funktion a ruft nach dem Setzen von x die Funktion b auf. Diese wiederum ruft nach dem Setzen von x die Funktion a auf. Die Methode `main` startet das Spiel durch einen Aufruf von a .

Führt man das Computerprogramm aus, kommt es zu einer unendlichen Rekursion. Durch den zyklischen, gegenseitigen Aufruf der Funktionen a und b endet das Programm nicht selbstständig. Ohne ein Abbrechen der Software von außen (unter Unix beispielsweise durch die Tastenkombination CTRL+C) läuft es endlos. Der Wert von x wird während der Laufzeit durch die Funktionen a und b abwechselnd auf die Werte 1 und 0 gesetzt. Nach einem Abbruch des Programms existiert die Variable x nicht mehr.

Beim Start des Spiels wird dem Laplaceschen Dämon folgende Frage gestellt:

- Welchen Wert hat die Variable x ?

Die nachfolgenden Kapitel beschäftigen sich mit der Beantwortung der Frage und erörtern, in wie weit der Laplacesche Dämon dazu in der Lage wäre.

3. Der Wert der Variablen x

Aus probabilistischer Sicht ist die Frage des vorangehenden Kapitels leicht zu beantworten:

- Die Variable x nimmt mit einer Wahrscheinlichkeit von 50% den Wert 1 und mit einer Wahrscheinlichkeit von 50% den Wert 0 an

⁵ Siehe (Gödel, 1931) und (Hofstadter, 1991)

⁶ Andere Programmiersprachen wie Java und R können das Spiel in dieser Form nicht ausführen. Bei beiden Sprachen führt die entstehende endlose Rekursion zu einem Stapelüberlauf (Stack Overflow). Die C++ Implementierung der GUN Compiler Collection (kurz: gcc) verursacht diesen Fehler nicht. Sie verfügt über eine sogenannte Tail-Recursion-Optimierung.

Geht man von einer unendlichen Laufzeit des Programms aus, sind die oben genannten Wahrscheinlichkeiten exakt. Wird die Software von außen abgebrochen, sind die Wahrscheinlichkeiten Näherungswerte.⁷

Aus deterministischer Sicht können bei einer unendlichen Laufzeit des Laplaceschen Spiels mindestens folgende Aussagen getroffen werden:

- Beim Start des Spiels hat die Variable x den Wert 0
- Wenn die Variable x derzeit den Wert 0 hat, wird sie anschließend den Wert 1 annehmen
- Wenn die Variable x derzeit den Wert 1 hat, wird sie anschließend den Wert 0 annehmen

Bei einer unendlichen Laufzeit des Laplaceschen Spiels kann der Wert von x nicht eindeutig festgelegt werden. Er nimmt sowohl den Wert 0, als auch den Wert 1 an. Gleiches gilt für eine endliche Laufzeit.⁸ Das nachfolgende Kapitel beschäftigt sich damit, ob x für einen bestimmten Zeitpunkt t deterministisch festgelegt werden kann.⁹

4. Wert der Variablen x zum Zeitpunkt t

Abbildung 2 stellt die Ergebnisse des Laplaceschen Spiels bei einer begrenzten Laufzeit (25 Runden) dar. Um eine Ausgabe zu erhalten, wurde das Programm zu Illustrationszwecken erweitert.¹⁰ Die Funktionen a und b geben in der erweiterten Version nach jedem Setzen der Variable x ihren aktuellen Wert aus. Erst danach rufen sie die jeweils andere Funktion auf. Zusätzlich zum Wert x schreiben sie die aktuelle Systemzeit in Mikrosekunden. So wird ersichtlich, wann die Variable x den jeweiligen Wert hatte.

Das nachfolgende Beispiel zeigt, dass die erste Ausgabe des erweiterten Programms in Mikrosekunde 1530607219337640 stattfand. Der Wert der Variable x war

⁷ Bei einer endlichen Laufzeit kann der Wert 0 einmal mehr vorkommen, als der Wert 1. Das Prinzip der großen Zahlen sorgt bei zunehmender Laufzeit für immer exaktere Ergebnisse. Das Spiel muss mindestens eine Runde ausgeführt werden.

⁸ Das Spiel muss mindestens eine Runde ausgeführt werden.

⁹ Die Analyse geht von einer absoluten Zeit aus, die für den Laplaceschen Dämon und für den ausführenden Rechner gleich schnell voranschreitet.

¹⁰ Die Erweiterung dient der Veranschaulichung. Dem Laplaceschen Dämon wird das in Kapitel 2 beschriebene Spiel vorgelegt.

zu diesem Zeitpunkt 1. Die nächste Meldung stammt aus Mikrosekunde 1530607219337733. Zu diesem Zeitpunkt hatte x den Wert 0. Durch die Betrachtung des Outputs wird ersichtlich, dass der Zeitabstand zwischen den Ausgaben schwankt. In den meisten Fällen liegt er (näherungsweise) bei einer Mikrosekunde. In einem Fall liegt er darüber, in einem Fall liegt er unter einer Mikrosekunde.

```
x = 1 1530607219337640
x = 0 1530607219337733
x = 1 1530607219337734
x = 0 1530607219337735
x = 1 1530607219337736
x = 0 1530607219337737
x = 1 1530607219337738
x = 0 1530607219337739
x = 1 1530607219337740
x = 0 1530607219337741
x = 1 1530607219337742
x = 0 1530607219337743
x = 1 1530607219337744
x = 0 1530607219337745
x = 1 1530607219337746
x = 0 1530607219337747
x = 1 1530607219337748
x = 0 1530607219337749
x = 1 1530607219337750
x = 0 1530607219337751
x = 1 1530607219337752
x = 0 1530607219337752
x = 1 1530607219337753
x = 0 1530607219337754
x = 1 1530607219337755
```

Abbildung 2: Werte der Variable x zum Zeitpunkt t bei 25 Läufen des Laplaceschen Spiels

Die Geschwindigkeit in der die Variable x ihren Wert ändert, hängt von verschiedenen Faktoren ab. Hierzu zählt die Leistung des Prozessors des Rechners, auf dem das Spiel ausgeführt wird. In der erweiterten Form des Spiels ist die Ausgabe der Ergebnisse ein wesentlicher Faktor.¹¹

Der Laplacesche Dämon kann *vergangene* Ergebnisse des Spiels nachvollziehen, indem er sich alle Werte von x mit dem jeweiligen Zeitstempel merkt. Für die Betrachtung der Variable in der *Gegenwart* und in der *Zukunft* spielt die Schnelligkeit der Ausführung des Spiels eine bedeutende Rolle.

Nehmen wir an, ein Computer A führt das Laplacesche Spiel aus. Er ist aufgrund seiner Rechengeschwindigkeit in der Lage den Wert der Variablen x jede Mikrosekunde zu wechseln. Der Laplacesche Dämon könnte einen Computer B nutzen, um den aktuellen Wert von x auszulesen und anschließend den nächsten Wert zum Zeitpunkt $t+1$ zu berechnen. Computer B muss die Vorhersage dazu jedoch in weniger als einer Mikrosekunde durchführen. Ansonsten handelt es sich nicht um eine Vorhersage, sondern um die Feststellung des Status Quo oder der Vergangenheit. Computer B muss in seiner Vorhersage dazu alle Faktoren berücksichtigen, die sich auf die Rechengeschwindigkeit von Computer A auswirken.

Nehmen wir an, der Laplacesche Dämon verfügt über Computer B. Kann er die Ergebnisse des Spiels vorhersagen? Führt er das Spiel auf Rechner A aus, ist dies eventuell möglich. Läuft das Spiel hingegen auf Computer B selbst, lautet die Antwort nein. Computer B kann nicht schneller rechnen, als er selbst.¹² Der Laplacesche Dämon benötigt in diesem Fall einen Computer C, der performanter ist, als Computer B. Führt er das Spiel auf Computer C aus, benötigt er einen Computer D etc.

Nehmen wir an, der Laplacesche Dämon verfügt über den ultimativen Supercomputer Z. Dieser ist in der Lage unendlich schnell zu rechnen. Führt er das Laplacesche Spiel mit dieser Maschine aus, kann er den Wert von x nicht deterministisch bestimmen. Der Wert von x wäre gleichzeitig 1 und 0. Es ist lediglich eine probabilistische Aussage möglich.

¹¹ Es ist davon auszugehen, dass das in Kapitel 2 dokumentierte Spiel wesentlich schneller läuft, als die erweiterte Version.

¹² Seth Lloyd zeigt dies in (Lloyd, 2012).

5. Fazit

Der Laplacesche Dämon kann den Wert der Variablen x des Laplaceschen Spiels zu einem bestimmten Zeitpunkt t nur mit einem Rechner vorhersagen, der schneller rechnet, als der Computer, der das Experiment ausführt. Fordert man den Laplaceschen Dämon auf, das Spiel mit seinem schnellsten Rechner zu spielen, ist er nicht in der Lage den Wert der Variablen x für jeden beliebigen Zeitpunkt t vorherzusagen. Dies widerspricht der Definition des Laplaceschen Dämons. Er ist damit falsifiziert. Die Aussage gilt im Besonderen für die Annahme, dass der Laplacesche Dämon selbst der schnellste Rechner der Welt ist. In diesem Fall könnte man ihn dazu auffordern das Laplacesche Spiel selbst zu spielen.

A. Praktische Relevanz

Das Laplacesche Spiel ist keine rein theoretische Überlegung. Das in Kapitel 2 beschriebene C++ Programm kann ein Computer ausführen. Sind die Hypothesen des Aufsatzes korrekt, ist er nicht in der Lage, die von ihm produzierten Werte von x für jeden beliebigen Zeitpunkt t selbst vorherzusagen. Voraussetzung ist, dass der Prozess des Laplaceschen Spiels mit der maximal möglichen Geschwindigkeit der Maschine ausgeführt und nicht verlangsamt wird.

Auch außerhalb der Informatik ist Rekursion ein beobachtbares Phänomen. Ein Beispiel ist der Aktienmarkt. Hier sind Händler aktiv, die mit ihrem Verhalten andere Händler beeinflussen. Deren Verhalten spiegelt wiederum auf die Verursacher zurück. So entsteht eine Art Laplacesches Spiel. Die Variable x kann man sich als den aktuellen Preis einer Aktie vorstellen. Sobald ein Händler eine Aktie kauft oder verkauft, beeinflusst er ihren Preis und damit das Verhalten der Marktteilnehmer.

Der Ökonom John Maynard Keynes veranschaulicht das rekursive Prinzip des Aktienmarktes mit Hilfe eines Schönheitswettbewerbs.¹³ Wettet ein Spekulant auf den Ausgang eines Schönheitswettbewerbs, ist es nicht die beste Strategie auf die Teilnehmerin (oder den Teilnehmer) zu setzen, die (oder den) er selbst am schönsten findet. Vielmehr sollte er auf die Teilnehmerin setzen, die eine Mehrzahl der Jury wählen wird. Die Jury einer Aktie sind alle Händler. Auf einem Markt kann es so

¹³ Vgl. (Keynes, 1936, S. 156)

zu einer Rekursion kommen, in der jeder Händler versucht das Verhalten der anderen Händler vorherzusehen. Dies schließt die Überlegung ein, welche Annahmen die anderen Händler über seine eigenen Annahmen treffen etc. Analog zum Laplaceschen Spiel lässt sich am Finanzmarkt ein Wettlauf der Rechengeschwindigkeiten beobachten. Im sogenannten „Algorithmic Trading“ werden Kauf- und Verkaufsaufträge heute im untersekundlichen Bereich ausgeführt. Eine deterministische Vorhersagbarkeit zukünftiger Aktienkurse ist für längere Zeiträume am Aktienmarkt nicht beobachtbar.

B. Referenzen

Ferber, R. (2003). *Philosophische Grundbegriffe*. München: C.H. Beck.

Gödel, K. (01. Dezember 1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, S. 173-198.

Heisenberg, W. (März 1927). Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift für Physik. Band 43, Nr. 3*, S. 172–198.

Hofstadter, D. R. (1991). *Gödel, Escher, Bach*. München: Klett-Cotta.

Keynes, J. M. (1936). *The General Theory Of Employment Interest And Money*. London: Macmillan.

Lloyd, S. (2012). A Turing test for free will. *Philosophical Transactions of the Royal Society A*, S. 3597–3610.

Silver, N. (2012). *The Signal and the Noise*. London: Penguin Random House.