

Algebraic Statistics

Benjamin C. W. Brown
B.Brown@ed.ac.uk

~

1st November 2020

Contents I

1. EXAMPLE
2. PENROSES THESE
3. PENROSES BEWEIS
4. KRITIK VON DARYL McCULLOUGH
5. PENROSES ANTWORT
6. EIN BISSCHEN CODE
 - 6.1 Größter gemeinsamer Teiler
 - 6.2 Ein Pseudocode Beispiel
 - 6.3 Literaturverzeichnis

formales System

Ein System welches Regeln enthält, mit deren Hilfe sich mathematische Aussagen beweisen lassen und mit denen aus bereits bewiesenen Aussagen neue Aussagen abgeleitet werden können.

widerspruchsfrei

- ▶ A Aussage
- ▶ T formales System

$$\neg \exists A : T \rightarrow A \wedge T \rightarrow \neg A$$

Gödelisierung

Berechenbare Zuweisung einer eindeutigen natürlichen Zahl für jedes Wort¹ einer formalen Sprache.

Turing-Maschine

- ▶ Schreib-Lese-Kopf
- ▶ unendlich langer Eingabe-/Ausgabeband
- ▶ endliche Menge an Eingabesymbolen und internen Zuständen
- ▶ Initial- und Haltezustand
- ▶ Zustandsübergangsfunktion $q_{\text{aktuell}}, a_{\text{eingabe}} \rightarrow q_{\text{folge}}, a_{\text{ausgabe}}, d_{\text{direction}}$
- ▶ Universelle Turing-Maschine benötigt zusätzlich die Maschinenbeschreibung als Eingabe

¹endliche Folgen von Symbolen aus dem Alphabet der formalen Sprache

Penroses These

- ▶ in der Mathematik hat unser Denkprozess die reinste Form
 - ▶ falls das Denken bloß die Ausführung einer Berechnung ist, dann sollte man dies am deutlichsten in der Mathematik sehen
 - ▶ erstaunlicherweise zeigt sich genau das Gegenteil
 - ▶ *Der menschliche Denkprozess ist nicht algorithmisierbar*
- ↪ nicht durch eine (universelle) Turing-Maschine simulierbar

Gödelscher Unvollständigkeitssatz

Kein formelles System aus widerspruchsfreien mathematischen Regeln kann jemals ausreichen, alle wahren Aussagen der klassischen Arithmetik zu beweisen. M. a. W. es gibt immer Aussagen in dem System die wahr aber unbeweisbar sind.

- ▶ Arithmetik für den Menschen durch Intuition und Verstand begreifbar, aber nicht durch ein formelles System
- ▶ Indirekter Beweis m. H. des *Unvollständigkeitssatzes* und des *Halteproblems*
- ▶ Annahme es gibt ein formales System welches das menschliche Denken beschreibt

- ▶ Penrose verwendet die Begriffe Berechnung und Algorithmus gleichbedeutend
- ▶ Algorithmus entspricht dem Verhalten einer Turing-Maschine
- ▶ kann nicht nur arithmetische sondern auch logische Operationen zur Ableitung neuer Aussagen verwenden
- ▶ zeigt 'visuellen Beweis' als Unterstützung seiner These, dass der menschliche Verstand nicht berechenbare Methoden verwendet

Fragestellung

Finde eine Summe von aufeinanderfolgenden Hexagonal-Zahlen, beginnend von 1, welche keine Kubik-Zahl² ergibt.

Reihe von Hexagonal-Zahlen

$$\text{hex}(n) = \begin{cases} \sum_{i=1}^n \text{hex}(i-1) + 6 * i, & \text{wenn } n > 0, n \in \mathbb{N} \\ 1, & \text{wenn } n = 0 \end{cases}$$

² $x^3, x \in \mathbb{N}$

- ▶ Folge von Hexagonal-Zahlen für $n \in \{1, 2, 3, 4\} = \{1, 7, 19, 37\}$
- ▶ Summen der aufeinanderfolgenden Hexagonalzahlen
 $\{8, 27, 64\} = \{2^3, 3^3, 4^3\}$ für $n \in \{2, 3, 4\}$

Penroses Beweis

- ▶ $C(n), n \in \mathbb{N}$
- ▶ Indizierung aller Algorithmen $C_0, C_1, C_2 \dots C_q$
 - ▷ Jeder Algorithmus ist durch eine Turing-Maschine ausführbar
 - ▷ Jede Turing-Maschine kann durch eine Zeichenkette beschrieben werden
 - ▷ Zeichenketten sind indizierbar (z.B. durch Gödelisierung)
- ▶ Angenommen es existiert ein Algorithmus A der ausgibt ob $C_q(n)$ hält
- ▶ A beinhaltet dabei alle Methoden mathematischer Beweisführung (um zu überprüfen das eine Berechnung nicht hält)
- ▶ Wenn $A(q, n)$ hält, dann hält $C_q(n)$ **nicht**
- ▶ A muss nicht immer entscheiden können das $C_q(n)$ hält, aber er liefert auch keine falschen Antworten, d.h. er ist *widerspruchsfrei*

Gödel-Turing Satz \mathcal{G}

- a) $q = n$: wenn $A(n, n)$ hält, dann hält $C_n(n)$ nicht
- b) $A(n, n) = C_k(n), k \in \{0, \dots, q\}$

A	0	1	...	k	...	n-1	n
C_0	haltet						
C_1		haltet					
...			haltet				
C_k	haltet	haltet	haltet	haltet	nicht haltend	haltet	haltet
...					haltet		
C_{n-1}						haltet	
C_n							haltet

- c) für $n = k$: $A(k, k) = C_k(k)$
- d) wenn $A(k, k)$ hält, dann hält $C_k(k)$ nicht

- ▶ Wenn A widerspruchsfrei ist, dann wissen wir das $C_k(k)$ nicht hält und somit wissen wir etwas was A nicht beweisen kann.
- A kann unseren Verstand nicht abbilden
- ▶ \mathcal{G} : Mathematiker nutzen keinen bekannten widerspruchsfreien Algorithmus für mathematische Beweise

Kritik von Daryl McCullough

- ▶ Penrose gibt kein direktes Argument für seine These
- ▶ er zeigt keine konkrete Aufgabe die ein Mensch ausführen kann aber ein Computer nicht (Folie 8)
- ▶ stattdessen indirekter Beweis unter der Annahme es existiert 'Programm' welches genauso intelligent wie ein Mensch ist

Penroses Argument

- ▶ Penroses Denkvermögen wird durch ein formales System F beschrieben
- ▶ dies bedeutet das jede in F beweisbare Aussage S für Penrose wahr ist (F widerspruchsfrei)
- ▶ S ist also ein Satz (Theorem) von F und umgekehrt
- ▶ Annahme: Penrose weiss das F seinen Verstand beschreibt
 \hookrightarrow dies zieht—gemäß Penrose—den Glaube in die Widerspruchsfreiheit von F nach sich
- ▶ Aufgrund des *Unvollständigkeitssatzes* ist $G(F)$ eine wahre Aussage aber kein Satz von F
- ▶ Da Penrose an die Widerspruchsfreiheit von F glaubt, muss er folgern das $G(F)$ wahr ist

Uneindeutigkeit in der Bedeutung von F

Mögliche Interpretationen des formalen Systems F :

1. F stellt das *angeborene Denkvermögen* dar
2. es stellt das Denkvermögen zu einem bestimmten *Zeitpunkt* dar, einschließlich des empirischen Wissens
3. F beinhaltet *jegliches* erreichbare Wissen, obgleich durch Schlussfolgerung oder Empirie

- ▶ Uneindeutigkeit von F entscheidend bei der Frage ob der Mathematiker weiss das sein Denkvermögen durch F beschrieben wird (z. B. falls diese Erkenntnis durch empirisches Wissen erzeugt wird)
- ▶ Penrose umgeht dieses Problem indem er ein F' einführt, was F sowie alles was sich aus dem Bewusstsein durch F beschrieben zu sein ableiten lässt, enthält

Für \mathcal{G} getroffene Annahmen

- a) Logisches Denken ist *widerspruchsfrei*
- b) a) ist *unzweifelhaft wahr*

- ▶ \mathcal{G} beweist *nicht*, dass das menschliche Denkvermögen nicht berechenbar ist, sondern, falls es algorithmisierbar ist, zeigt es:
 - a) das es widersprüchlich ist
 - b) das es unmöglich für uns ist, gleichzeitig unser Urteilsvermögen (F) zu kennen und zu wissen ob es widerspruchsfrei ist
- ▶ Penrose behauptet: ‘ Falls wir einen Algorithmus F kennen der unser Denkvermögen beschreibt, dann sind wir gezwungen dessen Widerspruchsfreiheit zu folgern. ’

Lässt sich \mathcal{G} auf Menschen anwenden?

- ▶ Penroses Argumente bedingen die Fähigkeit bestimmte *zweifelsfreie Wahrheiten* zu erkennen
- ▶ eine falsche Aussage darf nicht als *zweifelsfrei wahr* bewertet werden, diese Tatsache ist aber *widerlegbar*

Quick and Dirty (selbstbezogene Aussage)

- ▶ G : 'Dieser Satz ist keine Überzeugung von Roger Penrose'
- ▶ $G \in \text{Penrose} \rightarrow G : \text{falsch}$
- ▶ Umgekehrt, wenn Penroses Überzeugungen widerspruchsfrei sind, dann $G \notin \text{Penrose} \rightarrow G : \text{wahr}$.
- ⚡ Wenn Penrose glaubt seine Überzeugungen wären widerspruchsfrei, sind sie es tatsächlich nicht.

Formeller (Paradoxon)

$$\begin{aligned} F(x) &: \mathbb{N} \rightarrow \mathbb{N}, x \in \mathbb{N} \\ F(x) &= \begin{cases} G(x) + 1, & \text{wenn } x = \text{code}(G), G : \mathbb{N} \rightarrow \mathbb{N} \\ 0, & \text{sonst} \end{cases} \\ N &= \text{code}(F) \\ F(N) &\stackrel{!}{=} F(N) + 1 \\ &\hookrightarrow F(N) = 0 \end{aligned}$$

- ▶ Paradoxon zeigt das die Idee einer *eindeutigen Definition* selbst nicht *eindeutig* sein kann
- ▶ Gleichermäßen kann die Idee einer *zweifelsfreien Wahrheit* selbst nicht

Gedankenexperiment

- ▶ Gegeben seien zwei Tasten für die Antwortmöglichkeiten **Ja** und **Nein**
 - ▶ die Testperson soll die entsprechende Taste als Antwort auf eine Entscheidungsfrage betätigen
 - ▶ ‘Werden Sie die *Nein*-Taste betätigen?’
- Keine korrekte Antwort auf die Frage möglich
- ▶ die Testperson kann ihr Wissen nicht mitteilen und ist somit nicht widerspruchsfrei
 - ▶ Wenn man Schlüsse aus seiner eigenen Widerspruchsfreiheit ableitet führt dies zu einem Fehler

McCulloughs Fazit

- ▶ Penroses Argument das unser Denkvermögen nicht formalisierbar ist, ist (in gewissen Sinne) korrekt
 - ▶ kein Problem des formalen Systems sondern eine implizite Beschränkung unserer Fähigkeit über unser eigenes Denkvermögen Schlüsse zu folgern
 - ▶ bis zu dem Punkt an dem wir unser eigenes logisches Denken verstehen, können wir uns über dessen Widerspruchsfreiheit nicht sicher sein
 - ▶ Andersherum, bis zu dem Punkt an dem wir wissen das wir widerspruchsfrei sind, können wir unser logisches Denken nicht formalisieren
- ↪ Diese Einschränkung besitzt jedes System was in der Lage ist über sich selbst zu schlussfolgern

Penroses Antwort

Wiederholung

- ▶ Annahme: ‘die Gesamtheit aller Methoden der mathematischen Beweisführung können in einem (nicht notwendigerweise berechenbaren) widerspruchsfreien formalen System F gebündelt werden’
- a) Wenn ich erkenne das ich F wäre, dann müsste F widerspruchsfrei sein und insbesondere F' ($F + \text{‘Ich bin } F\text{’}$)
- b) Also müsste auch $G(F')$ wahr sein
- ⚡ b) Erkenntnis außerhalb der Fähigkeit von $F' \hookrightarrow$ somit kann ich nicht durch F beschrieben sein
- ▶ Penrose hat ‘stärkere’ Form des Beweises gezeigt (ohne die Annahme zu wissen von F beschrieben worden zu sein)

- ▶ MC kritisiert das sich G nicht nur auf 'berechenbare' F anwenden lässt
- ▶ Penrose zeigt genau dies in Abschnitt 7.9
- ▶ wichtiger noch, die Fähigkeiten von F (Schlüsse über sich selbst zu ziehen) müssen eingeschränkt sein, da ansonsten Widersprüche auftreten³
- ▶ MC benutzt eine solche paradoxe Beweisführung (siehe Folie 19)

³Einschränkung wird nicht konkret benannt

```
1  # greatest common denominator
2  def gcd(a, b):
3      """Calculate the Greatest Common Divisor of a and b.
4
5      Unless b==0, the result will have the same sign as b (so that when b
6      is divided by it, the result comes out positive).
7      """
8      while b:
9          a, b = b, a%b
10     return a
```

Listing: GCD in python3

Algorithm 1: How to write algorithms

Result: Write here the result

```
1 initialization;  
2 while While condition do  
3   instructions;  
4   if condition then  
5     instructions1;  
6     instructions2;  
7   else  
8     instructions3;  
9   end  
10 end
```

nested items

- ▶ item
 - ▷ subitem
 - ▷ another subitem
 - subsubitem one
 - second subsubitem
- ▶ this item ist alerted

an example block

This should clarify something.

an alert block

Attention!

Fragen?

Bibliographie I