# 09

PDF

# 1.

diagonales Argument nach Cantor.
Idee: Angenommen, die Potenzmenge ist abzählbar, $\implies$ Widerspruch.

**Beweis (durch Widerspruch):**

1. **Annahme:** Angenommen, die Potenzmenge $\mathcal{P}(\mathbb{N})$ der natürlichen Zahlen sei abzählbar. Das heißt, es existiert eine Aufzählung aller Teilmengen von $\mathbb{N}$:

$$A_1, A_2, A_3, \ldots$$

   wobei jede Teilmenge $A_i$ eine Untermenge von $\mathbb{N}$ ist. Wir versuchen nun, eine bestimmte Teilmenge $D \subseteq \mathbb{N}$ zu konstruieren, die in dieser Liste nicht auftauchen kann.
2. **Konstruktion der diagonalen Menge:** Definiere die Menge

$$D := \{n \in \mathbb{N} \mid n \notin A_n\}.$$

   Das heißt, für jedes $n$ schauen wir uns die $n$-te aufgezählte Teilmenge $A_n$ an. Wenn $n$ nicht in $A_n$ enthalten ist, dann fügen wir $n$ in unsere neue Menge $D$ ein. Dies ist eine wohldefinierte Konstruktion, denn für jedes natürliche $n$ entscheiden wir eindeutig, ob $n$ zu $D$ gehört oder nicht.
3. **Widerspruch:** Wir behaupten nun, dass $D$ nicht in der aufgestellten Liste $(A_1, A_2, A_3, \ldots)$ vorkommt. Angenommen, es gäbe ein $k \in \mathbb{N}$ mit $D = A_k$. Dann untersuchen wir das Element $k$. Es gibt zwei Fälle:

   - Falls $k \in D$, dann gilt per Definition von $D$, dass $k \notin A_k$. Aber wenn $D = A_k$ wäre, müsste $k$ in $A_k$ enthalten sein. Das ist ein Widerspruch.
   - Falls $k \notin D$, dann gilt per Definition von $D$, dass $k \in A_k$. Aber wenn $D = A_k$, müsste $k$ nicht in $A_k$ sein. Auch dies ist ein Widerspruch.

   In beiden Fällen entsteht ein Widerspruch, der beweist, dass unsere Annahme $D = A_k$ falsch ist. Da unsere Annahme, dass $\mathcal{P}(\mathbb{N})$ abzählbar ist, zur Konstruktion dieser Situation führte, haben wir einen Widerspruch herbeigeführt.
4. **Schlussfolgerung:** Da wir unter der Annahme der Abzählbarkeit zu einem Widerspruch gelangen, ist die Annahme falsch. Folglich ist die Potenzmenge der natürlichen Zahlen $\mathcal{P}(\mathbb{N})$ nicht abzählbar, das heißt sie ist überabzählbar.

**Ergebnis:** Die Potenzmenge der natürlichen Zahlen ist überabzählbar, wie durch Cantors Diagonalargument gezeigt wurde.

# 2.

**Idee der Reduktion:**
Wir wollen zeigen, dass das Leerheitsproblem für Turingmaschinen unentscheidbar ist, indem wir es auf das Halteproblem reduzieren. Das Halteproblem ist gegeben als:
Eingabe: DTM $M$ und Wort$w$
Frage: Hält$M$bei Eingabe $w$?

Wir wissen, dass das Halteproblem unentscheidbar ist. Wir nehmen an, wir hätten einen Entscheider für das Leerheitsproblem. Unter dieser Annahme werden wir zeigen, wie sich damit das Halteproblem entscheiden ließe, was ein Widerspruch ist. Somit kann das Leerheitsproblem nicht entscheidbar sein.

**Reduktionskonstruktion:**
Gegeben seien eine DTM $M$ und ein Eingabewort $w$, für die wir entscheiden wollen, ob $M$ bei Eingabe von $w$ hält. Wir bauen aus $M$ und $w$ eine neue DTM $M'$ wie folgt:

**Idee der Konstruktion von $M'$:**
Die Maschine $M'$ ignoriert ihre eigene Eingabe (beliebiges Wort $y$) zunächst und simuliert $M$ auf $w$:

1. $M'$ erhält eine Eingabe $y$ (dieses $y$ ist beliebig, also egal welches Wort, z. B. aus $\Sigma^*$).
2. $M'$ simuliert die Berechnung von $M$ auf $w$.
   - Falls $M$ bei der Eingabe $w$ hält, geht $M'$ in einen akzeptierenden Zustand über und akzeptiert die Eingabe $y$. (Das heißt, sobald $M$ auf$w$hält, akzeptiert $M'$ jede Eingabe.)
   - Falls $M$ bei der Eingabe $w$ nicht hält, so hält auch die Simulation von $M$ in $M'$ nie an. In diesem Fall wird $M'$ niemals akzeptieren, ganz gleich, welche Eingabe $y$ gewählt wurde.

Mit anderen Worten:

- Wenn $M$ bei $w$ hält, dann akzeptiert $M'$ jedes Wort. Das bedeutet: $L(M') = \Sigma^*$, insbesondere ist also $L(M') \neq \emptyset$.
- Wenn $M$ bei $w$ nicht hält, akzeptiert $M'$ kein Wort. Dann gilt: $L(M') = \emptyset$.

**Nutzung des angenommenen Entscheiders für das Leerheitsproblem:**
Angenommen, wir hätten einen Algorithmus (einen Entscheider) für das Leerheitsproblem, der für jede DTM $M'$ entscheidet, ob $L(M') = \emptyset$ gilt. Dann könnten wir wie folgt vorgehen, um das Halteproblem zu entscheiden:

1. Wir konstruieren $M'$ aus $M$ und $w$ nach obiger Vorschrift.
2. Wir fragen den Leerheits-Entscheider: "Ist $L(M') = \emptyset$ ?"
   - Antwortet er "Ja" (leer), so wissen wir, dass $M$ bei $w$ nicht hält.
   - Antwortet er "Nein" (nicht leer), so wissen wir, dass $M$ bei $w$ hält.

Damit hätten wir eine Entscheidungsprozedur für das Halteproblem gefunden, was im Widerspruch dazu steht, dass das Halteproblem unentscheidbar ist.

**Schlussfolgerung:**
Unsere Annahme, dass das Leerheitsproblem entscheidbar ist, führt zu einem Widerspruch. Folglich ist das Leerheitsproblem (die Frage, ob die von einer DTM akzeptierte Sprache leer ist) unentscheidbar.

# 3.

**Idee der Reduktion:**

Wir wollen zeigen, dass das Nullstellenproblem unentscheidbar ist, indem wir es aus dem Halteproblem herleiten. Das Halteproblem ist wie folgt definiert:

- **Eingabe:** Eine deterministische Turingmaschine $M$ und eine Eingabe $w \in \Sigma^*$.
- **Frage:** Hält $M$ bei Eingabe $w$ an?

Wir wissen, dass das Halteproblem unentscheidbar ist. Nun nehmen wir an, das Nullstellenproblem sei entscheidbar. Das Nullstellenproblem ist wie folgt gegeben:

- **Eingabe:** Eine deterministische Turingmaschine $M'$ und ein Wort $u \in \Sigma^*$. Diese definieren eine totale berechenbare Funktion $f_{M'}(u)$ (z.B. eine ganzzahlige Funktion, die von $u$ abhängt).
- **Frage:** Gilt $f_{M'}(u) = 0$?

Wir wollen nun mittels einer Reduktion vom Halteproblem auf das Nullstellenproblem zeigen, dass letzteres nicht entscheidbar sein kann. Das heißt, wir nehmen an, wir hätten einen Entscheider für das Nullstellenproblem zur Verfügung, und zeigen dann, dass wir damit das Halteproblem entscheiden können. Da das Halteproblem unentscheidbar ist, muss diese Annahme zum Widerspruch führen.

**Konstruktion der Reduktion:**

Gegeben seien $M$ und $w$ aus dem Halteproblem. Wir wollen eine Maschine $M'$ und eine Eingabe $u$ konstruieren, sodass:

- Falls $M$ auf $w$ hält, dann ist $f_{M'}(u) = 0$.
- Falls $M$ auf $w$ nicht hält, dann ist $f_{M'}(u) \neq 0$ (oder niemals 0).

Auf diese Weise können wir durch Lösen des Nullstellenproblems für $(M', u)$ entscheiden, ob $M$ auf $w$ hält.

**Konstruktion von $M'$:**

1. **Eingabe von $M'$**: Die Maschine $M'$ bekommt eine Eingabe $u$ (diese kann beispielsweise eine leere Eingabe sein, da $u$ in dieser Konstruktion keine Rolle spielt).

2. **Simulation von $M$ auf $w$**: Bei der Berechnung von $f_{M'}(u)$ lässt $M'$ zunächst die Maschine $M$ auf Eingabe $w$ laufen.
3. **Fallunterscheidung:**
   - **Fall 1:** $M$ hält auf $w$. In diesem Fall lässt $M'$ die Berechnung enden und gibt als Ergebnis (Wert der Funktion $f_{M'}(u)$) die Zahl 0 aus.
   - **Fall 2:** $M$ hält auf $w$ nicht. In diesem Fall läuft die Simulation endlos weiter. Das heißt, $M'$ findet nie ein Halteereignis von $M$. Da $f_{M'}$ eine totale Funktion sein soll (das Nullstellenproblem nimmt an, dass $f_{M'}$ berechenbar und total ist), können wir an dieser Stelle wie folgt vorgehen:
     Wir konstruieren $M'$ so, dass sie nach einer bestimmten Codierung vorgeht. Zum Beispiel kann $M'$ folgende Funktion berechnen:

     $$f_{M'}(u) = \begin{cases} 0 & \text{wenn } M \text{ auf } w \text{ hält} \\ 1 & \text{wenn } M \text{ auf } w \text{ nicht hält} \end{cases}$$

     Da wir eine totale Funktion benötigen, muss $M'$ immer ein Ergebnis liefern. Der Trick ist: Wir machen die Berechnung von $M'$ folgendermaßen:

     - $M'$ simuliere $M$ auf $w$ für $n = 1, 2, 3, \ldots$ Schritte.
     - Bricht $M$ innerhalb von $n$ Schritten ab, gibt $M'$ sofort 0 aus.
     - Falls nach $n$ Schritten keine Halteaktion von $M$ gesehen wird, erhöhe $n$ um 1 und probiere es erneut.

     Da $M$ nie hält (im zweiten Fall), wird $M'$ diese Schleife endlos durchlaufen. Nun muss $f_{M'}$ dennoch total sein, d.h. einen numerischen Wert liefern. Dies scheint zunächst ein Problem, aber beachten Sie: Eine Funktion ist berechenbar, wenn es ein Programm gibt, das ihre Ausgabe bestimmt. "Total" bedeutet, dass die Funktion für jede Eingabe ein Ergebnis hat, aber nicht notwendigerweise endlich berechnet werden muss, um 0 oder 1 auszugeben.

     Hier liegt der eigentliche Widerspruch: Wenn wir davon ausgehen, dass uns das Nullstellenproblem eine Entscheidung (ja/nein) liefern kann, müsste es erkennen, ob $f_{M'}(u) = 0$ ist. Ist $M$ nicht haltend auf $w$, würde $M'$ jedoch niemals 0 ausgeben, sondern unendlich suchen. In der Praxis dient dieser Widerspruch zur Herleitung, dass kein solcher Entscheider existieren kann.

     Alternativ kann man die Funktion$f_{M'}$auch so konstruieren, dass sie eine leicht berechenbare Abbildung ist, deren Wert exakt dann 0 ist, wenn die Haltefrage bejaht wird, und sonst eindeutig von 0 verschieden ist (z.B. 1). Das Entscheidende ist, dass, um den Wert festzustellen, implizit das Halteproblem gelöst werden müsste.

**Schlussfolgerung:**

Hätten wir einen Entscheider für das Nullstellenproblem, könnten wir also folgendermaßen vorgehen, um das Halteproblem zu lösen:

- Gegeben $M, w$ (Instanz des Halteproblems), konstruiere $M'$ und$u$wie oben beschrieben.
- Frage den Nullstellen-Entscheider: „Ist $f_{M'}(u) = 0$?“.
  - Falls ja, so hält $M$ auf $w$.
  - Falls nein, so hält $M$ auf $w$ nicht.

Damit hätten wir einen Entscheider für das Halteproblem gebaut. Da das Halteproblem aber bekanntlich unentscheidbar ist, muss unsere Ausgangsannahme, dass das Nullstellenproblem entscheidbar ist, falsch sein. Also ist das Nullstellenproblem ebenfalls unentscheidbar.

---

# Übung

## 1.

$M$ abzählbar $:\iff \exists f : \mathbb{N} \to M.\ f \text{ bijektiv}$