# WAS IST WICHTIG?

## 1. Entscheidbarkeit und Berechenbarkeit

1. Eine Sprache (oder ein Problem) $L$ is **entscheidbar** genau dann, wenn es eine stets haltende Turingmaschine (bzw. ein stets haltendes Programm) $M$ gibt, so dass

$$\forall w(w \in L \Leftrightarrow M \text{ akzeptiert } w).$$

Eine Funktion $f$ heisst **berechenbar** genau dann, wenn es eine stets haltende Turingmaschine (bzw. ein stets haltendes Programm) $M$ gibt, so dass

$$\forall w( \ M \text{ berechnet Ausgabe } f(w) \text{ für Eingabe } w \ ).$$

2. Die Klasse entscheidbarer Probleme ist riesig. Insbesondere:
   - Alle Probleme in $\mathcal{P}$ oder $\mathcal{NP}$ sind entscheidbar. Warum gilt dies für $\mathcal{NP}$? Die Klasse $\mathcal{NP}$ wird doch durch nichtdeterministische Turingmaschinen definiert, während deterministische Turingmaschinen für die Definition entscheidbarer Probleme benutzt wird!
   - Alle Probleme, die durch Programme entschieden werden, die nur polynomiell großen Platz benötigen, also alle Probleme in $\mathcal{PSPACE}$, sind natürlich ebenfalls entscheidbar. Beachte, dass $\mathcal{NP} \subseteq \mathcal{PSPACE}$: Warum?
   - Wir hätten Entscheidbarkeit genauso gut mit probabilistischen Turingmaschinen definieren können, ohne die Klasse entscheidbarer Probleme zu ändern. Warum?
   - Auch Quantenberechnungen bringen nicht mehr!
3. All dies unterstützt die **Churchsche These**, dass nämlich unsere Definition der Entscheidbarkeit den Begriff *stets haltender Rechenverfahren* formalisiert.

## 2. Unentscheidbarkeit

1. Die Klasse entscheidbarer Probleme ist abzählbar, die Klasse unentscheidbarer Probleme hingegen überabzählbar (warum?). Glücklicherweise sind die meisten Probleme aber uninteressant.
2. Mit der Diagonalsprache **D** haben wir ein erstes, konkretes unentscheidbares Problem kennengelernt. Wie ist $D$ definiert und warum ist $D$ unentscheidbar?
3. Wir haben weitere unentscheidbare Probleme mit Hilfe von **Reduktionen** erhalten:

   Problem $L_1$ kann auf Problem $L_2$ (Notation: $L_1 \leq L_2$) reduziert werden, wenn es ein stets haltendes Programm $T$ (mit Ausgabe $T(w)$ für Eingabe $w$) gibt, so dass

$$\forall w(w \in L_1 \Leftrightarrow T(w) \in L_2)$$

   gilt.

   Hier sind die wesentlichen Eigenschaften der Reduktion:
   - Wenn $L_1 \leq L_2$ und $L_2 \leq L_3$, dann auch $L_1 \leq L_3$.
   - Angenommen, $L_1 \leq L_2$. Wenn $L_2$ entscheidbar, dann auch $L_1$. In äquivalenter Formulierung: Wenn $L_1$ unentscheidbar, dann auch $L_2$.

4. Wir haben $\overline{D} \leq \overline{U}$, sowie $U \leq H$ und $H \leq H_\varepsilon$ gezeigt. Als Konsequenz ist die universelle Sprache $U$, das Halteproblem und das spezielle Halteproblem unentscheidbar. (Warum gilt $H \leq H_\varepsilon$?)
5. Der **Satz von Rice** besagt, dass die Frage, ob ein Programm eine nicht-triviale Eigenschaft besitzt, unentscheidbar ist.
   – Wie wird der Satz von Rice genau formuliert?
   – Wie zeigt man mit dem Satz von Rice, dass die Frage, ob ein Programm immer hält, unentscheidbar ist?

## 3. Rekursiv aufzählbare Probleme

1. Ein Problem $L$ is **rekursiv aufzählbar** genau dann, wenn es ein Programm $M$ gibt mit
$$\forall w (w \in L \Leftrightarrow M \text{ akzeptiert } w).$$
Im Unterschied zur Entscheidbarkeit fordern wir also nicht, dass $M$ stets hält: Wenn $w \notin L$, dann darf $M$ nicht akzeptieren, muss also entweder verwerfen oder nicht halten!
2. Beispiele rekursiv aufzählbarer Sprachen:
   – $U, H$ und $H_\varepsilon$ sind rekursiv aufzählbar. Ist die Diagonalsprache rekursiv aufzählbar?
   – Wenn $G$ eine Grammatik ist, dann ist die Sprache $L(G)$ aller von $G$ erzeugten Worte rekursiv aufzählbar. Überraschenderweise kann sogar jede rekursiv aufzählbare Sprache von einer Grammatik erzeugt werden!
   Jede rekursiv aufzählbare Sprache ist eine formale Sprache und umgekehrt.
   – Die Menge aller aus einem endlichen Axiomensystem ableitbaren Formeln ist rekursiv aufzählbar (warum?). Gleiches gilt sogar, wenn das Axiomensystem rekursiv aufzählbar ist.
2. Wenn $L$ und $\overline{L}$ rekursiv aufzählbar sind, dann ist $L$ entscheidbar (warum?). Als Konsequenz folgt, dass die Komplemente der universellen Sprache, des Halteproblems oder des speziellen Halteproblems **nicht** rekursiv aufzählbar sind.
3. Wir haben **Gödels Unvollständigkeitssatz** gezeigt: Es gibt kein rekursiv aufzählbares Axiomensystem, so dass alle wahren Aussagen der „Zahlentheorie“ ableitbar sind.

## 4. Komplexitätsklassen

In Komplexitätsklassen werden alle Probleme gesammelt, die mit einem bestimmten Ressourcenaufwand gelöst werden können. Wichtige Klassen, in der Reihenfolge ihrer Berechnungskraft:

1. $\mathcal{P}$, die Klasse aller durch deterministische Programme in polynomieller Zeit lösbaren Probleme. Zu $\mathcal{P}$ gehören alle in der Algorithmentheorie behandelten Probleme:
   – Graph-theoretische Probleme wie das Zusammenhangsproblem, das kürzeste-Wege Problem oder das minimale Spannbaumproblem (warum?),
   – Probleme der Bioinformatik wie etwa das paarweise Alignment oder die Voraussage der RNA-Sekundärstruktur.
   – Eines der schwierigsten Probleme in $\mathcal{P}$ ist die lineare Programmierung.
2. $\mathcal{NP}$, die Klasse aller durch nichtdeterministische Programme in polynomieller Zeit lösbaren Probleme. Zu $\mathcal{NP}$, aber wahrscheinlich nicht zu $\mathcal{P}$ gehören alle $\mathcal{NP}$-vollständigen Probleme wie etwa
   – das Erfüllbarkeitproblem,

  - viele graph-theoretische Probleme wie das Clique Problem, das Vertex-Cover oder das Färbungsproblem
  - und die 0-1 Programmierung oder die ganzzahlige Programmierung als $\mathcal{NP}$-vollständige Varianten der linearen Programmierung.
3. $\mathcal{PSPACE}$, die Klasse aller durch deterministische Programme in polynomiellem Platz lösbaren Probleme. Zu $\mathcal{PSPACE}$, aber wahrscheinlich nicht zu $\mathcal{NP}$ gehören
  - QBF, die wahren quantifizierten Booleschen Formeln (warum?)
  - und die Bestimmung eines Gewinnzugs in vielen 2-Personen Spielen wie etwa das Geographie-Spiel.
4. Die Klasse aller entscheidbaren Probleme beinhaltet zum Beispiel die Presburger Arithmentik, die nicht in $\mathcal{PSPACE}$ liegt. In der Presburger Arithmetik befinden sich alle Aussagen (mit der Addition als Relation), die in den natürlichen Zahlen wahr sind.
5. Die Klasse der rekursiv aufzählbaren Probleme und
6. die Klasser aller restlichen unentscheidbaren Probleme. (Welche Probleme, die nicht Komplemente rekursiv aufzählbarer Probleme sind, gehören hierzu?)