4.1 Unentscheidbare Probleme

Ziel: Viele (natürliche) Probleme sind nicht entscheidbar, aber wenigstens semi-entscheidbar.

Dazu nehmen wir in diesem Kapitel für alle Turingmaschinen $M = (Q, \Sigma, \Gamma, \iota, \delta, \Box, F)$ an:

- $Q = \{1^n \mid 1 \leq n \leq N\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, 2, 3\}$, d.h. $3 = \Box$
- $\iota = 1$, $F = \{1^n \mid f \leq n \leq N\}$

Dies ist keine (wesentliche) Einschränkung, denn jede Turing-Maschine kann in eine „äquivalente“ umgewandelt werden, die diese Eigenschaften hat.

Codes/Namen der Turing-Maschine M

1. Seien $1 \leq m, n \leq N$, $a, b \in \Gamma$ und $x \in \{-1, 0, 1\}$. Der Code von $(1^m, a, 1^n, b, x)$ ist das Wort
$$001^m 01^{a+1} 01^n 01^{b+1} 01^{2+x}\,.$$

2. jede Verkettung aller Codes von $(1^m, a, 1^n, b, x)$ mit $(1^n, b, x) \in \delta(1^m, a)$ ist ein Code von δ

3. jedes Wort $w0001^f$, wobei w ein Code von δ ist, ist ein Code von M. Sei $L_{TM} \subseteq \{0, 1\}^*$ die Menge aller Codes von TM. Ist $w \in L_{TM}$ ein Code der TM M, so setze $M_w = M$.

Bemerkung. Die Sprache L_{TM} ist entscheidbar, d.h. es gibt eine TM M_{TM}, die sich bei Eingabe von $w \in \{0, 1\}^*$ wie folgt verhält:

- Gilt $w \in L_{TM}$, so hält M_{TM} in einem akzeptierenden Zustand.

- Gilt $w \notin L_{TM}$, so hält M_{TM} in einem nicht-akzeptierenden Zustand.

Insbes. hält die TM M_{TM} bei jeder Eingabe.

Lemma 4.1. *Das spezielle Halteproblem*

$$K = \{w \in \{0,1\}^* \mid w \in L_{TM},\ M_w \textit{ hält bei Eingabe von } w\}$$

ist semi-entscheidbar.

BewIdee:
Wir werden eine TM S konstruieren, die sich bei Eingabe eines Wortes w wie folgt verhält:

- Gilt $w \in L_{TM}$ und hält die TM M_w bei Eingabe von w irgendwann, so akzeptiert S das Wort w (d.h. hält in einem akzeptierenden Zustand).
- Ansonsten hält S bei Eingabe von w nicht.

1. Vortest: die TM S verhält sich zunächst wie die TM M_{TM}. Wenn M_{TM} in einem nicht-akzeptierenden Zustand hält, so geht S in eine Endlosschleife über. Wenn M_{TM} in einem akzeptierenden Zustand hält, so rechnet S wie folgt weiter (auf dem Band steht weiterhin nur das Wort w).

2. Initialisierung: die TM S schreibt hinter die Eingabe w das Wort $0000100002w$
Idee: die simulierte TM M_w ist im Initialzustand 1^1 und auf ihrem Band steht das Wort w und der Kopf der simulierten TM steht auf der ersten Position von w

3. Simulieren: in jeder „Runde“ sucht die TM S im Wort w den Code $001^p01^{a+1}01^q01^{b+1}01^{2+x}$ einer Anweisung $(q, b, x) \in \delta(p, a)$, wobei p der augenblickliche Zustand der simulierten Maschine M_w und a das von der simulierten Maschine M_w augenblicklich gelesene Zeichen ist.
Dann ersetzt die TM S den Zustand p durch q (dabei muß u.U. viel hin-und-her geschoben werden), das gelesene Zeichen a durch b und bewegt den simulierten Kopf (d.h. die 2) um x.
Ist $p \in F$ und existiert kein Code der Form $001^p01^{a+1}01^q01^{b+1}01^{2+x}$, so wechselt die TM S in einen akzeptierenden Zustand und hält an. Ansonsten geht S in die nächste Runde.
□

Lemma 4.2. *Das spezielle Halteproblem K ist nicht entscheidbar.*

BewIdee:
Angenommen, E wäre eine TM, die K entscheidet, d.h. E hält bei jeder Eingabe und $L(E) = K$.
Wir konstruieren aus E eine neue TM M:

- Wenn E akzeptiert (d.h. in einem akzeptierenden Zustand anhält), so bewegt M ihren Kopf immer weiter nach rechts (insbes. hält M nicht an).
- Wenn E in einem nicht-akzeptierenden Zustand anhält, so geht M in einen Finalzustand und hält an.

Sei w ein Code von M, d.h. $w \in L_{TM}$ mit $M_w = M$. Dann gilt:

M hält bei Eingabe von w gdw. E akzeptiert w nicht
gdw. M_w hält bei Eingabe von w nicht,
im Widerspruch zu $M = M_w$. □

Folgerung. *Das spezielle Halteproblem K ist semi-entscheidbar, sein Komplement $\Sigma^* \setminus K$ ist nicht semi-entscheidbar.*

BewIdee:
Da K nach Lemma 4.1 semi-entscheidbar ist, existiert eine TM S mit $L(S) = K$.

Wäre $\Sigma^* \setminus K$ semi-entscheibar, so gäbe es eine TM $\overline{S}$ mit $L(\overline{S}) = \Sigma^* \setminus K$.

Dann könnte man eine TM M konstruieren, die S und $\overline{S}$ „parallel" ausführt.

Diese TM M entscheidet K, was nach Lemma 4.2 nicht möglich ist. $\square$

Diese Beweisidee kann man verallgemeinern zu einem Beweis von:

Satz. *Sei $A \subseteq \Sigma^*$. Dann sind äquivalent*

- *A ist entscheidbar.*
- *A und $\Sigma^* \setminus A$ sind beide semi-entscheidbar.*

Satz 4.3. *Das Halteproblem*

$$H = \{w\#x \mid w \in L_{TM}, M_w \textit{ hält bei Eingabe von } x\}$$

ist nicht entscheidbar.

BewIdee:
Angenommen, H wäre entscheidbar. Dann gäbe es eine TM M mit $L(M) = H$, die bei jeder Eingabe anhält.

Wir konstruieren hieraus eine neue TM M', die zunächst hinter ihre Eingabe w das Wort $\#w$ schreibt und dann die TM M startet.

Die TM M' hält bei jeder Eingabe.

Sei $w \in L_{TM}$. Dann gilt

$$\begin{array}{lll} w \in L(M') & \text{gdw.} & w\#w \in H \\ & \text{gdw.} & M_w \text{ hält bei Eingabe von } w \\ & \text{gdw.} & w \in K, \end{array}$$

d.h. $L(M') = K$ und M' hält bei jeder Eingabe, im Widerspruch zu Lemma 4.2. Also ist das Halteproblem H nicht entscheidbar. □

Der Beweis von Satz 4.3 verwendet die folgende Idee:

(1) Wir konstruieren eine TM F, die aus einer Eingabe w für K eine Eingabe $f(w) = w\#w$ für H berechnet und zeigen, daß $w \in K \iff f(w) \in H$ gilt.

Eine solche Funktion f heißt *Reduktion von K auf H*. Sie muß

- von einer TM berechnet werden können und
- $w \in K \iff f(w) \in H$ erfüllen.

Wir schreiben $K \leq H$ für „es gibt eine Reduktion von K auf H“.
Der Beweis geht dann wie folgt weiter:

(2) Angenommen, H wäre entscheidbar. Dann gäbe es eien TM M, die H entscheidet. Die TM M' führt dann die TM F u‘nd M nacheinander aus und entscheidet K.

Da aber K nicht entscheidbar ist, kann eine solche TM M' nicht existieren. Also ist H nicht entscheidbar.

Allgemeiner erhalten wir

Satz 4.4. *Seien $A, B \subseteq \Sigma^*$ mit $A \leq B$.*

- *Ist B entscheidbar, so auch A (d.h. ist A unentscheidbar, so auch B).*
- *Ist B semi-entscheidbar, so auch A (d.h. ist A nicht semi-entscheidbar, so auch B).*

Hiermit kann man viele weitere Unentscheidbarkeiten zeigen:

Satz. *Die folgenden Probleme sind nicht entscheidbar:*

1. *Das* Halteproblem bei leerer Eingabe

$$H_0 = \{w \in \{0,1\}^* \mid w \in L_{TM} \Longrightarrow M_w \textit{ hält bei Eingabe von } \varepsilon\}$$

(analog für jede andere Eingabe v)

2. *Das* Regularitätsproblem

$$\{w \in \{0,1\}^* \mid w \in L_{TM} \Longrightarrow L(M_w) \textit{ ist regulär}\}$$

*allgemeiner gilt der **Satz von Rice.***
Sei $\mathcal{P}$ eine Eigenschaft von Sprachen und seien N und P TM, so daß $L(P)$ $\mathcal{P}$ erfüllt und $L(N)$ nicht. Dann ist

$$\{w \in \{0,1\}^* \mid w \in L_{TM} \Longrightarrow L(M_w) \text{ erfüllt } \mathcal{P}\}$$

nicht entscheidbar.

3. *Das Disjunktheitsproblem, das Inklusionsproblem, das Äquivalenzproblem und das Regularitätsproblem für kontextfreie Sprachen sind ebenfalls nicht entscheidbar (unter jeder „sinnvollen“ Kodierung von kontextfreien Grammatiken durch Wörter über $\{0,1\}$).*

4. *Die Menge der SQL-Anfragen, die in jeder Datenbankinstanz die leere Antwort erzeugen.*

5. *Das Äquivalenzproblem für Programme in Ihrer Lieblingssprache (wobei wir einen unendlichen Speicher voraussetzen).*

Zusammenfassung

- Die Church-Turing These erlaubt es zu untersuchen, inwiefern Probleme durch Algorithmen lösbar oder nicht lösbar sind.
- Jedes entscheidbare Problem ist semi-entscheidbar, aber nicht umgekehrt.
- Es gibt Problem, die nicht semi-entscheidbar (und damit nicht entscheidbar) sind.
- Reduktionen erlauben es, die Unentscheidbarkeit eines Problems aus der Unentscheidbarkeit eines anderen herzuleiten.
- Sehr viele natürliche Probleme sind nicht entscheidbar, oft aber wenigstens semi-entscheidbar.