Teil III: Berechenbarkeit

§11. Turingmaschinen

§12. Zusammenhang zwischen Turingmaschinen und Grammatiken

§13. Berechenbarkeit, Entscheidbarkeit, Aufzählbarkeit und Zusammenhänge

§14. Primitiv rekursive Funktionen und LOOP-Programme

§15. μ-rekursive Funktionen und WHILE-Programme

§16. Universelle Turingmaschinen und unentscheidbare Probleme

§17. Weitere unentscheidbare Probleme

Teil IV: Komplexität

§18. Komplexitätsklassen

§19. NP-vollständige Probleme

§ 13. Berechenbarkeit, Entscheidbarkeit, Aufzählbarkeit

Wir führen die Grundbegriffe der Theorie der Berechenbarkeit ein und setzen sie zueinander in Beziehung

Neben dem Berechnen von Funktionen betrachten wir dabei auch das Entscheiden von Relationen, was für viele Informatik-Probleme natürlich ist.

Wir werden hier – gemäß der Church-Turing These – Turingmaschinen als Berechnungsmodell verwenden

Definition 13.1 (Berechenbarkeit von Funktionen)

Eine partielle oder totale Funktion

$$f : (\Sigma^*)^n \to \Sigma^*.$$

heißt berechenbar falls sie Turing-berechenbar ist.

Zur Erinnerung:

Bei partiellen Funktionen entspricht Undefiniertheit des Funktionswertes der Nichtterminierung des Berechnungsverfahren (der DTM) bei dieser Eingabe.

Berechenbare totale Funktionen nennt man auch rekursiv.

Berechenbare partielle Funktionen nennt man auch partiell rekursiv.

In der Informatik erfordern viele Probleme nur eine ja/nein-Antwort anstelle der Berechnung eines “echten” Funktionswertes:

- Leerheitsproblem, z.B. für NEAs: gegeben NEA $\mathcal{A}$, ist $L(\mathcal{A}) = \emptyset$?
- Wortproblem, z.B. für kontextfreie Grammatiken:
  gegeben kontextfreie Grammatik G und Wort w, ist $w \in L(G)$?
- etc

Derartige Probleme nennen wir Entscheidungsprobleme.

Entscheidungsprobleme formalisieren wir nicht als Funktion, sondern als Relation.

Eine n-stellige Relation $R : (\Sigma^*)^n$ ist eine Menge von n-Tupeln $(w_1, \ldots, w_n)$ von Wörtern über Σ

Das assoziierte ja/nein-Problem ist:
ist ein gegebenes n-Tupel $(w_1, \ldots, w_n)$ in der Relation R oder nicht?

Am Beispiel des Äquivalenzproblems für kontextfreie Grammatiken:

1. Jede Grammatik $G = (N, \Sigma, P, S)$ kann als Wort

$$\mathrm{code}(G) \in \Gamma^*$$

über einem (festen!) Alphabet Γ aufgefaßt werden.

2. Das Äquivalenzproblem für Typ 1-Sprachen ist dann die binäre Relation

$$R = \{(\mathrm{code}(G_1), \mathrm{code}(G_2)) \mid G_1, G_2 \text{ kontextsensitiv}, L(G_1) = L(G_2)\} \subseteq \Gamma^* \times \Gamma^*.$$

Definition 13.2 (Entscheidbarkeit, partielle Entscheidbarkeit, rekursive Aufzählbarkeit von Relationen)

Eine n-stellige Relation $R \subseteq (\Sigma^*)^n$

1. heißt entscheidbar, falls es DTM gibt, die bei Eingabe von $(w_1, \ldots, w_n) \in (\Sigma^*)^n$

   - in akzeptierender Stoppkonf. anhält wenn $(w_1, \ldots, w_n) \in R$
   - in nicht-akzeptierender Stoppkonf. anhält wenn $(w_1, \ldots, w_n) \notin R$

   Die Eingabe $(w_1, \ldots, w_n)$ entspricht dabei wieder der Start-konfiguration $q_0 w_1 \not{b}\, w_2 \not{b} \cdots \not{b}\, w_n$

Wir verwenden DTMs ohne Beschränkung der Allgemeinheit (Satz 11.4) in Analogie mit der Berechenbarkeit von Funktionen

Entscheidbarkeitsbeweise haben wir im Prinzip schon viele geführt

vor kurzem z.B. für das Wortproblem für kontextsensitive Grammatiken

Dabei haben wir bisher nur nie

- Turingmaschinen benutzt sondern einen intuitiven Berechenbarkeitsbegriff (Church-Turing-These)
- das Problem als Relation über einer Menge von Wörtern dargestellt

Beachte:

Kann als Spezialfall der Berechnung totaler Funktionen betrachtet werden:

Eine Relation R ist entscheidbar genau dann wenn die totale Funktion

$$\chi_R(w_1, \ldots, w_n) := \begin{cases} a & \text{wenn } (w_1, \ldots, w_n) \in R \\ \varepsilon & \text{sonst} \end{cases}$$

(ihre charakteristische Funktion) berechenbar ist.

Definition 13.2 (Entscheidbarkeit, partielle Entscheidbarkeit, rekursive Aufzählbarkeit von Relationen)

2. R heißt partiell entscheidbar, falls es DTM gibt, die bei Eingabe von $(w_1, \ldots, w_n) \in (\Sigma^*)^n$

   - terminiert, falls $(w_1, \ldots, w_n) \in R$ ist und
   - nicht terminiert, falls $(w_1, \ldots, w_n) \notin R$ ist.

Beachte:
R ist also partiell entscheidbar, falls R der Definitionsbereich einer berechenbaren partiellen Funktion f ist

(denn beide Begriffe sind über DTMs definiert und jede DTM berechnet partielle Funktion und bezeugt partielle Entscheidbarkeit einer Relation)

Beispiel:

Das Wortproblem für Typ 0-Grammatiken ist partiell entscheidbar

Verwende folgende Variante des Algorithmus für Typ 1-Sprachen:

setze $M := \{S\}$
repeat
  for all $u \in M$ und $v \in \Sigma^*$ mit $u \vdash_G v$ do
    setze $M := M \cup \{v\}$
  if $w \in M$ then
    halte an
forever

Offensichtlich leicht als DTM implementierbar

Wir werden später sehen, dass dieses Problem nicht entscheidbar ist.

Bemerkung 13.3

Für Mengen (1-stellige Relationen) R stimmen die Begriffe „partiell entscheidbar“ und „Turing-erkennbar“ überein.

Unterschiede:

- Turing-erkennbar fordert bei $w \in R$ anhalten in Endzustand
- partiell-entscheidbar fordert bei $w \notin R$ nicht-Terminierung

Turing-erkennbar $\Rightarrow$ partiell entscheidbar

Gehe von allen nicht-akzeptierenden Stoppkonfigurationen aus in Endlosschleife

partiell entscheidbar $\Rightarrow$ Turing-erkennbar

Gehe von allen Stoppkonfigurationen aus in einen Endzustand

Definition 13.2 (Entscheidbarkeit, partielle Entscheidbarkeit, rekursive Aufzählbarkeit von Relationen)

3. R heißt rekursiv aufzählbar, falls R von einer Aufzähl-Turingmaschine aufgezählt wird. Diese ist wie folgt definiert:

   Eine Aufzähl-Turingmaschine $\mathcal{A}$ ist eine DTM, die einen speziellen Ausgabezustand $q_{Ausgabe}$ hat.

   Eine Ausgabekonfiguration für eine n-stellige Relation ist von der Form

   $$uq_{Ausgabe}\, w_1 \cancel{b}\, w_2 \cancel{b} \ldots w_n \cancel{b} v$$

   mit $u, v \in \Gamma^*$ und $w_1, \ldots, w_n \in \Sigma^*$.

   Diese Konfiguration erzeugt die Ausgabe $(w_1, \ldots, w_n)$.

   Die durch $\mathcal{A}$ aufgezählte Relation ist

   $$\begin{aligned} R \;=\; \{(w_1, \ldots, w_n) \in (\Sigma^*)^n \mid\ & (w_1, \ldots, w_n) \\ & \text{erzeugt durch eine Ausgabekonfiguration,} \\ & \text{die von } \mathcal{A} \text{ ausgehend von } q_0 \cancel{b} \text{ erreicht wird}\}. \end{aligned}$$

Beispiel:

Für jede Typ 0-Grammatik G ist die Menge $L(G)$ rekursiv aufzählbar

Verwende folgende Variante des vorigen Verfahrens:

> setze $M := \{S\}$
> repeat
>   for all $u \in M$ und $v \in \Sigma^*$ mit $u \vdash_G v$ do
>   setze $M := M \cup \{v\}$
>   if v is Terminalwort then
>   gehe in $q_{Ausgabe}$ und gib v aus
> forever

Die ähnlichen Beispiele für partielle Entscheidbarkeit und rekursive Aufzählbarkeit sind kein Zufall

Satz 13.4

Es sei $R \subseteq (\Sigma^*)^n$ eine Relation.

1. R ist rekursiv aufzählbar gdw R ist partiell entscheidbar.
2. Ist R entscheidbar, so auch partiell entscheidbar.
3. Ist R entscheidbar, so ist auch das Komplement $\overline{R} = (\Sigma^*)^n \setminus R$ entscheidbar
4. R ist entscheidbar gdw R und $\overline{R}$ partiell entscheidbar sind.

R rekursiv aufzählbar $\Rightarrow$ R partiell entscheidbar:

Es sei $\mathcal{A}$ eine Aufzähl-DTM für R.

Die Maschine $\mathcal{A}'$ arbeitet wie folgt:

- Sie speichert die Eingabe $(w_1, \ldots, w_n)$ auf zusätzlichem Band.
- Sie beginnt mit der Aufzählung von R.
- Bei jeder Ausgabekonfiguration überprüft sie, ob die entsprechende Ausgabe mit der gespeicherten Eingabe übereinstimmt.

  Wenn ja, so terminiert $\mathcal{A}'$.

  Sonst sucht sie die nächste Ausgabekonfiguration von $\mathcal{A}$.
- Terminiert $\mathcal{A}$, ohne daß die Eingabe $(w_1, \ldots, w_n)$ ausgegeben wurde, so geht $\mathcal{A}'$ in Endlosschleife.

$\mathcal{A}'$ terminiert also genau dann, wenn $(w_1, \ldots, w_n)$ in der Aufzählung vorkommt.

R partiell entscheidbar $\Rightarrow$ R rekursiv aufzählbar:

Es sei $\mathcal{A}$ ein partielles Entscheidungsverfahren für R, d.h.
$\mathcal{A}$ terminiert auf Eingabe $(w_1, \ldots, w_n)$ gdw. $(w_1, \ldots, w_n) \in R$

Sei $\vec{w}^{(1)}, \vec{w}^{(2)}, \vec{w}^{(3)}, \ldots$ eine "geeignete" Aufzählung von $(\Sigma^*)^n$.

Idee für die Maschine $\mathcal{A}'$:

(1) Führe einen Schritt der Berechnung von $\mathcal{A}$ auf Eingabe $\vec{w}^{(1)}$ aus.

(2) Führe zwei Schritte die Berechnung von $\mathcal{A}$ auf Eingaben $\vec{w}^{(1)}$ und auf Eingabe $\vec{w}^{(2)}$ aus.

$\vdots$

(n) Führe n Schritte die Berechnung von $\mathcal{A}$ auf den Eingaben $\vec{w}^{(1)}, \ldots, \vec{w}^{(n)}$ aus.

$\vdots$

Terminiert $\mathcal{A}$ für eine dieser Eingaben, so gebe die Eingabe aus und mache weiter.

Beachte:

Man kann nicht $\mathcal{A}$ zunächst ganz mit Eingabe $\vec{x}^{(1)}$ laufen lassen (ohne Schrittbeschränkung), da $\mathcal{A}$ darauf nicht terminieren muß.

Einige Details:

- Die konkrete Ordnung $\vec{w}^{(1)}, \vec{w}^{(2)}, \vec{w}^{(3)}, \ldots$ ergibt sich einfach aus einem geeigneten Verfahren zum Aufzählen
- Auf Band 2 realisiert $\mathcal{A}'$ einen Zähler $i = 1, 2, 3, \ldots$ für die Schrittzahl
- Auf Band 3 realsiert $\mathcal{A}'$ einen Zähler j für die aktuelle Eingabe
- Auf Band 4 erzeugt $\mathcal{A}'$ $\vec{w}^{(j)}$ durch Aufzählung $\vec{w}^{(1)}, \vec{w}^{(2)}, \ldots, \vec{w}^{(j)}$
- Auf Band 5 führt $\mathcal{A}'$ die ersten i Schritte der Berechnung von $\mathcal{A}$ auf $\vec{w}^{(j)}$ aus, zählt dabei auf Band 6 die schon gemachten Schritte
- Band 1 fungiert als Ausgabeband

2. Ist R entscheidbar, so auch partiell entscheidbar.

   Eine DTM $\mathcal{A}$, die R entscheidet, kann wie folgt modifiziert werden:

   - hält $\mathcal{A}$ in nicht-akzeptierender Stoppkonfiguration, so gehe in Endlosschleife.
   - keine Änderung, wenn $\mathcal{A}$ in akzeptierender Stoppkonfiguration stoppt

3. Ist R entscheidbar, so auch das Komplement $\overline{R} = (\Sigma^*)^n \setminus R$.

   Eine DTM $\mathcal{A}$, die R entscheidet, wird wie folgt zu einer DTM für $\overline{R}$ modifiziert:

   - $F = Q \setminus F$ (tausche Endzustände und nicht-Endzustände)

4. R entscheidbar $\Rightarrow$ R und $\overline{R}$ partiell entscheidbar:

ergibt sich unmittelbar aus 2) und 3).

R und $\overline{R}$ partiell entscheidbar $\Rightarrow$ R entscheidbar:

Sind R und $\overline{R}$ partiell entscheidbar, so mit 1) auch rekursiv aufzählbar.

Für Eingabe $\vec{x}$ läßt man die Aufzähl-DTMs $\mathcal{A}$ und $\mathcal{B}$ für R und $\overline{R}$ parallel laufen,

d.h. jeweils abwechselnd ein Schritt von $\mathcal{A}$ auf einem Band gefolgt von einem Schritt von $\mathcal{B}$ auf dem anderen.

Die Eingabe $\vec{x}$ kommt in einer der beiden Aufzählungen vor:

$$\vec{x} \in (\Sigma^*)^n = R \cup \overline{R}.$$

- Kommt $\vec{x}$ bei $\mathcal{A}$ vor, so stoppe in Endzustand
- sonst , so stoppe in nicht-Endzustand