Grundlagen der Theoretischen Informatik

Till Mossakowski

Fakultät für Informatik
Otto-von-Guericke Universität
Magdeburg

Wintersemester 2014/15

Grammatiken und Turing-Maschinen

Beispiel: Sei $G = (V, \Sigma, R, S)$ mit $V = \{S, A, D, \triangleright, \heartsuit\}$, $\Sigma = \{a\}$ und

$$\begin{aligned} R = \{ \quad S &\to \triangleright A \heartsuit, \\ \triangleright &\to \triangleright D, \\ D\heartsuit &\to \heartsuit, \\ DA &\to AAD, \\ \triangleright &\to \varepsilon, \\ \heartsuit &\to \varepsilon, \\ A &\to a \quad \} \end{aligned}$$

$$L(G) = \{a^{2^n} \mid n \geq 0\}$$

Satz:
Sei G eine Grammatik. Dann ist $L(G)$ rekursiv aufzählbar.

Beweisskizze: Sei $G = (V, \Sigma, R, S)$ eine Grammatik. Wir simulieren Ableitungen von G mit einer nichtdeterministischen 2-Band-Turing-Maschine M, so dass $L(G) = L(M)$:

Band 1: $w \in \Sigma^*$
Band 2: $x \in (V \cup \Sigma)^*$ mit $S \Rightarrow_G^* x$ (anfangs $x = S$)

Die Turing-Maschine arbeitet in Simulationsphasen. Zu Beginn jeder Phase wählt die Turing-Maschine eine von $|R|+1$ Optionen nichtdeterministisch aus: Eine von $|R|$ Regeln auswählen oder Ableiten beenden.

Regel $u \to v$ anwenden: Startposition für Suche nach u auf Band 2 nichtdeterministisch auswählen, Bandinhalt ab Startposition mit u vergleichen. Falls Übereinstimmung gefunden wurde, zugehörigen Teil von Band 2 löschen, Platz für v anpassen und v auf Band 2 schreiben.

Ableiten beenden: Inhalt der Bänder 1 und 2 vergleichen. Falls Übereinstimmung gefunden, zu q_{accept} wechseln.

Falls in einem der Schritte keine Übereinstimmung gefunden wurde, geht die Turing-Maschine in eine Endlosschleife über.

M hält bei Eingabe w genau dann wenn $w \in L(G)$. ■

Satz:
Sei L eine rekursiv aufzählbare Sprache. Dann gibt es eine Grammatik G mit $L = L(G)$.

Beweisskizze:
Sei $M = (K, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}})$ eine Turing-Maschine. O.B.d.A. dürfen wir annehmen, dass M den akzeptierenden Haltezustand q_{accept} nur in Konfiguration $(\varepsilon, q_{\text{accept}}, \varepsilon)$ erreicht, d.h., die Turing-Maschine M leert stets das Band, bevor sie in den akzeptierenden Haltezustand wechselt.

Ferner dürfen wir o.B.d.A. annehmen, dass $K \cap \Gamma = \emptyset$.

Wir konstruieren eine Grammatik $G = (V, \Gamma, R, S)$ mit $V = K \cup \{S, \triangleright, \triangleleft\}$. Die Beweisidee ist, Konfigurationen von M als Wörter über $V \cup \Gamma$ darzustellen und Berechnungen von M rückwärts durch Ableitungen zu simulieren.

Eine Konfiguration

$$(w_1, q, w_2)$$

entspricht dem Ableitungswort

$$\triangleright w_1 q w_2 \triangleleft$$

Um die Produktionsregeln nicht unnötig kompliziert zu machen, lassen wir in Ableitungswörtern Blanksymbole nach $\triangleright$, das das linke Ende markiert, zu, ebenso vor $\triangleleft$, welches das rechte Ende markiert.

Wir tragen nun dafür Sorge, dass sich Übergänge von M in Produktionsregeln widerspiegeln.

Für jeden Übergang $\delta(q,a) = (p,b,S)$ fügen wir $pb \to_G qa$ zu R hinzu.

Für jeden Übergang $\delta(q,a) = (p,b,L)$ fügen wir für alle $\sigma \in \Gamma$ die Produktionsregel $p\sigma b \to_G \sigma qa$ zu R hinzu.

Für jeden Übergang $\delta(q,a) = (p,b,R)$ fügen wir $bp \to_G qa$ zu R hinzu.

Gegebenenfalls ist es nötig, vor Anwendung einer Produktionsregeln Blanksymbole vor $\triangleleft$ bzw. hinter $\triangleright$ zu erzeugen. Deshalb fügen wir die Regeln $\triangleright \to_G \triangleright\sqcup$ und $\triangleleft \to_G \sqcup\triangleleft$ zu R hinzu.

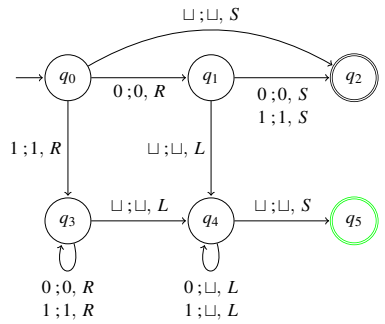
Um überflüssige Blanksymbole entfernen zu können, fügen wir die Regeln $\triangleright\sqcup \to_G \triangleright$ und $\sqcup\triangleleft \to_G \triangleleft$ zu R hinzu.

Aus dem Startsymbol kann mit der Produktionsregel $S \to_G \triangleright q_{\text{accept}} \triangleleft$ ein Wort erzeugt werden, das der einzigen akzeptierenden Konfiguration entspricht.

Um aus Wörtern, die Startkonfigurationen entsprechen, das Eingabewort extrahieren zu können, fügen wir die Produktionsregeln $\triangleright s \to_G \varepsilon$ und $\triangleleft \to_G \varepsilon$ zu R hinzu.

Wir unterbrechen den Beweis für ein Beispiel.

$M = (\{q_0,\ldots,q_5\},\{0,1\},\{0,1,\sqcup\},\delta,q_0,q_5,q_2)$:



$G = (V,\Sigma,R,S)$:

$$\begin{aligned} V &= \{S,\triangleright,\triangleleft,q_0,\ldots,q_5\} \\ \Sigma &= \{0,1,\sqcup\} \end{aligned}$$

R enthält folgende Produktionsregeln:

$$\begin{aligned} S &\to_G \triangleright q_5 \triangleleft \\ \triangleright &\to_G \triangleright\sqcup \\ \triangleleft &\to_G \sqcup\triangleleft \\ \sqcup\triangleleft &\to_G \triangleleft \\ \triangleright\sqcup &\to_G \triangleright \\ \triangleright q_0 &\to_G \varepsilon \\ \triangleleft &\to_G \varepsilon \end{aligned}$$

$$\begin{aligned} 0q_1 &\to_G q_0 0 \\ 1q_3 &\to_G q_0 1 \\ 0q_3 &\to_G q_3 0 \\ 1q_3 &\to_G q_3 1 \\ q_4\sigma\sqcup &\to_G \sigma q_3\sqcup \quad \text{für alle } \sigma\in\Sigma \\ q_4\sigma\sqcup &\to_G \sigma q_4 0 \quad \text{für alle } \sigma\in\Sigma \\ q_4\sigma\sqcup &\to_G \sigma q_4 1 \quad \text{für alle } \sigma\in\Sigma \\ q_4\sigma\sqcup &\to_G \sigma q_1\sqcup \quad \text{für alle } \sigma\in\Sigma \\ q_2 0 &\to_G q_1 0 \\ q_2 1 &\to_G q_1 1 \\ q_5\sqcup &\to_G q_4\sqcup \\ q_2\sqcup &\to_G q_0\sqcup \end{aligned}$$

$$\begin{aligned} \delta(q_0,0) &= (q_1,0,R) \\ \delta(q_0,1) &= (q_3,1,R) \\ \delta(q_3,0) &= (q_3,0,R) \\ \delta(q_3,1) &= (q_3,1,R) \\ \delta(q_3,\sqcup) &= (q_4,\sqcup,L) \\ \delta(q_4,0) &= (q_4,\sqcup,L) \\ \delta(q_4,1) &= (q_4,\sqcup,L) \\ \delta(q_1,\sqcup) &= (q_4,\sqcup,L) \\ \delta(q_1,0) &= (q_2,0,S) \\ \delta(q_1,1) &= (q_2,1,S) \\ \delta(q_4,\sqcup) &= (q_5,\sqcup,S) \\ \delta(q_0,\sqcup) &= (q_2,\sqcup,S) \end{aligned}$$

		100	$\triangleright q_0 \to_G \varepsilon, \triangleleft \to_G \varepsilon$
	$(\varepsilon,q_0,100)$	$\overset{*}{}_G\Leftarrow$ $\triangleright q_0 100\triangleleft$	$1q_3 \to_G q_0 1$
$\vdash_M$	$(1,q_3,00)$	$_G\Leftarrow$ $\triangleright 1q_3 00\triangleleft$	$0q_3 \to_G q_3 0$
$\vdash_M$	$(10,q_3,0)$	$_G\Leftarrow$ $\triangleright 10q_3 0\triangleleft$	$0q_3 \to_G q_3 0$
$\vdash_M$	$(100,q_3,\varepsilon)$	$_G\Leftarrow$ $\triangleright 100q_3\triangleleft$	$\sqcup\triangleleft \to_G \triangleleft$
		$_G\Leftarrow$ $\triangleright 100q_3\sqcup\triangleleft$	$q_4 0\sqcup \to_G 0q_3\sqcup$
$\vdash_M$	$(10,q_4,0)$	$_G\Leftarrow$ $\triangleright 10q_4 0\sqcup\triangleleft$	$\triangleleft \to_G \sqcup\triangleleft$
		$_G\Leftarrow$ $\triangleright 10q_4 0\triangleleft$	$q_4 0\sqcup \to_G 0q_4 0$
$\vdash_M$	$(1,q_4,0)$	$_G\Leftarrow$ $\triangleright 1q_4 0\sqcup\triangleleft$	$\triangleleft \to_G \sqcup\triangleleft$
		$_G\Leftarrow$ $\triangleright 1q_4 0\triangleleft$	$q_4 1\sqcup \to_G 1q_4 0$
$\vdash_M$	$(\varepsilon,q_4,1)$	$_G\Leftarrow$ $\triangleright q_4 1\sqcup\triangleleft$	$\triangleright\sqcup \to_G \triangleright, \triangleleft \to_G \sqcup\triangleleft$
		$\overset{*}{}_G\Leftarrow$ $\triangleright\sqcup q_4 1\triangleleft$	$q_4\sqcup\sqcup \to_G \sqcup q_4 1$
$\vdash_M$	$(\varepsilon,q_4,\varepsilon)$	$_G\Leftarrow$ $\triangleright q_4\sqcup\sqcup\triangleleft$	$\triangleleft \to_G \sqcup\triangleleft$
		$_G\Leftarrow$ $\triangleright q_4\sqcup\triangleleft$	$q_5\sqcup \to_G q_4\sqcup$
$\vdash_M$	$(\varepsilon,q_5,\varepsilon)$	$_G\Leftarrow$ $\triangleright q_5\sqcup\triangleleft$	$\triangleleft \to_G \sqcup\triangleleft$
		$_G\Leftarrow$ $\triangleright q_5\triangleleft$	$S \to_G \triangleright q_5\triangleleft$
		$_G\Leftarrow$ S	

Wir fahren nun mit der Beweisskizze fort.

Der zu beweisende Satz folgt aus folgendem

Lemma:
Seien (w_1,q,w_2) und (w_3,p,w_4) Konfigurationen. Falls $(w_1,q,w_2)\vdash_M (w_3,p,w_4)$ so gilt $\triangleright w_3 p w_4\triangleleft \Rightarrow^*_G \triangleright w_1 q_1 w_2\triangleleft$. Umgekehrt folgt aus $\triangleright w_3 p w_4\triangleleft \Rightarrow^*_G \triangleright w_1 q_1 w_2\triangleleft$, dass $(w_1,q,w_2)\vdash^*_M (w_3,p,w_4)$ gilt.

Nun gilt

$$\begin{aligned} & w\in L(G) \\ \text{g.d.w.}\quad & S\Rightarrow_G \triangleright q_{\text{accept}}\triangleleft \Rightarrow^*_G \triangleright s w\triangleleft \Rightarrow^*_G w \\ \text{g.d.w.}\quad & \triangleright q_{\text{accept}}\triangleleft \Rightarrow^*_G \triangleright s w\triangleleft \\ \text{g.d.w.}\quad & (\varepsilon,s,w)\vdash^*_M (\varepsilon,q_{\text{accept}},\varepsilon) \\ \text{g.d.w.}\quad & w\in L(M) \end{aligned}$$

■

3

Berechenbarkeitstheorie

Berechenbarkeit

Definition:
Seien Σ_1 und Σ_2 Alphabete. Eine Funktion $f:\Sigma_1^*\to\Sigma_2^*$ heißt Turing-berechenbar (oder einfach berechenbar*), falls es eine Turing-Maschine $M=(K,\Sigma_1,\Gamma,\delta,s,h,q_{\text{reject}})$ gibt, so dass für alle $w\in\Sigma_1^*$ gilt*

$$(\varepsilon,s,w)\vdash^*_M (\varepsilon,h,f(w))$$

Turing-Maschinen und reale Computer

Turing-Maschinen können die Berechnungen realer Computer simulieren, auch die paralleler Supercomputer. Von dem Problem, dass die Resourcen realer Computer beschränkt sind, mal abgesehen, können reale Computer andererseits Turing-Maschinen simulieren.

Entscheidbarkeit

0?

1?

Gödelisierung

Um mit Objekten im Zusammenhang mit Sprachen umgehen zu können, müssen wir Objekte als Wörter über einem Alphabet kodieren.

Insbesondere wollen wir jeder Turing-Maschine ein Wort über einem Alphabet, d.h., eine Zahl, zuordnen. Dabei sollen verschiedenen Turing-Maschinen verschiedene Wörter zugeordnet werden, d.h., die Zuordnung muss injektiv sein.

Eine Turing-Maschine $(K, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}})$ können wir beispielsweise so kodieren:

Seien

$$f : \{0, \ldots, |K| - 1\} \to K$$

und

$$g : \{0, \ldots, |\Gamma| - 1\} \to \Gamma$$

Bijektionen mit $f(0) = s$, $f(1) = q_{\text{accept}}$, $f(2) = q_{\text{reject}}$ und $g(0) = \sqcup$.

Ferner sei

$$h : \{0, 1, 2\} \to \{L, R, S\}$$

mit $h(0) = L$, $h(1) = R$ und $h(2) = S$.

$$\begin{aligned} q \in K &\mapsto \langle q \rangle = 01^{f^{-1}(q)+1}0 \\ \sigma \in \Gamma &\mapsto \langle \sigma \rangle = 01^{g^{-1}(\sigma)+1}0 \\ D \in \{L, R, S\} &\mapsto \langle D \rangle = 01^{h^{-1}(D)+1}0 \\ \delta(q, a) = (p, b, D) &\mapsto \langle q \rangle \langle a \rangle \langle p \rangle \langle b \rangle \langle D \rangle \end{aligned}$$

M wird nun als Konkatenation der Kodierungen der Übergänge kodiert.

Ein Eingabewort wird als Konkatenation der Kodierungen der einzelnen Symbole kodiert.

Mit der beschriebenen Kodierung gilt

Lemma:
Die Sprache

$$L = \{w \in \{0,1\}^* \mid w \textit{ ist die Kodierung einer Turing-Maschine}\}$$

ist entscheidbar.

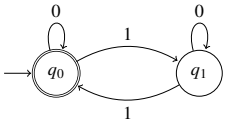
Analog zur beschriebenen Kodierung von Turing-Maschinen können wir auch deterministischen und nichtdeterministischen endlichen Automaten Wörter aus $\{0,1\}^*$ zuordnen, die die Automaten kodieren. Einen deterministischen endlichen Automaten $M = (K, \Sigma, \Delta, s, F)$ können wir beispielsweise so kodieren:

$$\begin{aligned} q \in K &\mapsto \langle q \rangle = 01^{f^{-1}(q)+1}0 \\ \sigma \in \Sigma &\mapsto \langle \sigma \rangle = 01^{g^{-1}(\sigma)+1}0 \\ (q, a, p) \in \Delta &\mapsto \langle q \rangle \langle a \rangle \langle p \rangle \end{aligned}$$

f sei so, dass es ein k gibt, so dass $F = \{q \in K \mid f^{-1}(q) \geq k\}$.

Falls ferner $f(j) = s$, so kodieren wir M nun durch $1^j 001^k 00$ gefolgt von der Konkatenation der Kodierungen der Elemente von Δ.

Beispiel:



100100011001001100110011001001001001001001100110

$$f(0) = q_1 \quad f(1) = q_0 \qquad g(0) = 0 \quad g(1) = 1$$

Analog zur beschriebenen Kodierung von Turing-Maschinen und endlichen Automaten können wir auch Kellerautomaten als Wörter aus $\{0,1\}^*$ kodieren.

Ebenso können wir Grammatiken als Wörter aus $\{0,1\}^*$ kodieren, indem wir die Elemente von $V \cup \Sigma$ durchnummerieren und Grammatiken eineindeutig als Folge der Kodierungen ihrer Produktionsregeln kodieren, wobei eine Produktionsregel $u \to_G v$ kodiert wird durch

$$00\langle u\rangle 00\langle v\rangle 00$$

Auf ähnliche Weise können wir reguläre Ausdrücke kodieren.

Mit den beschriebenen Kodierungen gilt

Lemma:
Die Sprachen

$\{w \in \{0,1\}^* \mid w$ *ist die Kodierung eines deterministischen endlichen Automaten*$\}$,

$\{w \in \{0,1\}^* \mid w$ *ist die Kodierung eines nichtdeterministischen endlichen Automaten*$\}$,

$\{w \in \{0,1\}^* \mid w$ *ist die Kodierung eines Kellerautomaten*$\}$,

$\{w \in \{0,1\}^* \mid w$ *ist die Kodierung einer Grammatik*$\}$ *und*

$\{w \in \{0,1\}^* \mid w$ *ist die Kodierung eines regulären Ausdrucks*$\}$

sind entscheidbar.

Universelle Turing-Maschine

Eine *Universelle Turing-Maschine* erhält als Eingabe eine Beschreibung einer Turing-Maschine $\langle M\rangle$ und eine zugehörige kodierte Eingabe $\langle w\rangle$. Die Universelle Turing-Maschine U simuliert dann die Berechnung von M bei Eingabe w.

Eine Universelle Turing-Maschine kann leicht als 3-Band Turing-Maschine realisiert werden:

Band 1: kodierter Bandinhalt der simulierten Maschine
Band 2: Kodierung der simulierten Maschine
Band 3: kodierter Zustand der simulierten Maschine

Entscheidbare Probleme bei formalen Sprachen

Sei $\mathscr{A}_{\mathsf{DFA}} = \{\langle M,w\rangle \mid M$ ist ein deterministischer endlicher Automat, der w akzeptiert$\}$.

Satz: *Die Sprache $\mathscr{A}_{DFA}$ ist entscheidbar.*

Sei $\mathscr{A}_{\mathsf{NFA}} = \{\langle M,w\rangle \mid M$ ist ein nichtdeterministischer endlicher Automat, der w akzeptiert$\}$.

Satz: *Die Sprache $\mathscr{A}_{NFA}$ ist entscheidbar.*

Sei $\mathscr{A}_{\mathsf{REX}} = \{\langle R,w\rangle \mid R$ ist ein regulärer Ausdruck, der w erzeugt$\}$.

Satz: *Die Sprache $\mathscr{A}_{REX}$ ist entscheidbar.*

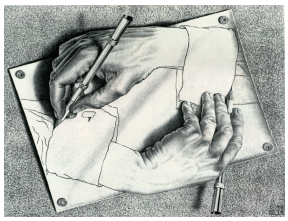
Sei $\mathscr{A}_{\mathsf{CFG}} = \{\langle G,w\rangle \mid G$ ist eine kontextfreie Grammatik, die w erzeugt$\}$.

Satz: *Die Sprache $\mathscr{A}_{CFG}$ ist entscheidbar.*

Sei $\mathscr{A}_{\mathsf{PDA}} = \{\langle M,w\rangle \mid M$ ist ein Kellerautomat, der w akzeptiert$\}$.

Satz: *Die Sprache $\mathscr{A}_{PDA}$ ist entscheidbar.*

Unentscheidbarkeit



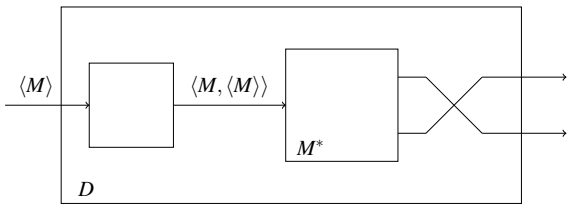
```
#include <stdio.h>
char *s="include <stdio.h>%cchar *s=%c%s%c;%cint main(){printf(s,10,34,s,34,10,10);}%c";
int main(){printf(s,10,34,s,34,10,10);}
```


Sei $\mathscr{A}_{\mathsf{TM}} = \{\langle M,w\rangle \mid M$ ist eine Turing-Maschine, die w akzeptiert$\}$.

Satz: *Die Sprache $\mathscr{A}_{TM}$ ist nicht entscheidbar.*

Beweis:
Nehmen wir an, es gäbe eine Turing-Maschine M^*, die die Sprache $\mathscr{A}_{\mathsf{TM}}$ entscheidet.

Basierend auf M^* könnten wir dann eine Turing-Maschine D konstruieren, die bei Eingabe $\langle M\rangle$ Turing-Maschine M^* für Eingabe $\langle M,\langle M\rangle\rangle$ simuliert, aber im verwerfenden Haltezustand hält, falls M^* diese Eingabe akzeptiert und im akzeptierenden Haltezustand sonst.



D verwirft $\langle M\rangle$ genau dann wenn M die Eingabe $\langle M\rangle$ akzeptiert. Was tut D bei Eingabe $\langle D\rangle$? Die Turing-Maschine D verwirft $\langle D\rangle$ genau dann wenn D die Eingabe $\langle D\rangle$ akzeptiert!

Damit ist die Annahme, dass M^* existiert, ad absurdum geführt. ■

Halteproblem

$\mathscr{H} = \{\langle M, w\rangle \mid$ Turingmaschine M hält bei Eingabe $w\}$

Satz: *Die Sprache $\mathscr{H}$ ist nicht entscheidbar.*

Beweis:
Nehmen wir an, es gäbe eine Turing-Maschine H, die $\mathscr{H}$ entscheidet. Dann könnten wir uns das zu Nutze machen, um eine Turing-Maschine M^* zu konstruieren, die $\mathscr{A}_{\mathsf{TM}}$ entscheidet:

Bei Eingabe $\langle M, w\rangle$ führt M^* zunächst die Berechnung von H für diese Eingabe aus.

Falls H die Eingabe akzeptiert, so wissen wir, dass M bei Eingabe w hält. M^* simuliert dann die Berechung von M bei Eingabe w und hält im akzeptierenden Haltezustand, falls M Eingabe w akzeptiert, und im verwerfenden Haltezustand sonst.

Falls H die Eingabe verwirft, so wissen wir, dass M bei Eingabe w nicht hält. M^* hält im verwerfenden Haltezustand.

M^* würde die nicht entscheidbare Sprache $\mathscr{A}_{\mathsf{TM}}$ entscheiden. Also kann H nicht existieren. Also ist das Halteproblem $\mathscr{H}$ unentscheidbar. ■

Entscheidbarkeit und rekursive Aufzählbarkeit

Satz:
Eine Sprache $L \subset \Sigma^*$ ist entscheidbar genau dann wenn sowohl L als auch ihr Komplement $\overline{L} = \Sigma^* - L$ rekursiv aufzählbar sind.

Beweis:
Sei $M = (K, \Sigma, \Gamma, \delta, s, q_{\text{accept}}, q_{\text{reject}})$ eine Turing-Maschine, die L entscheidet. Dann ist M auch eine Turing-Maschine, die L akzeptiert und $\overline{M} = (K, \Sigma, \Gamma, \delta, s, q_{\text{reject}}, q_{\text{accept}})$, also die zu M bis auf das Vertauschen der Haltezustände identische Turing-Maschine, akzeptiert $\overline{L}$.

Falls M eine Turing-Maschine ist, die L akzeptiert und M' eine Turing-Maschine, die $\overline{L}$ akzeptiert, so können wir daraus eine 2-Band-Turing-Maschine $\tilde{M}$ konstruieren, die L entscheidet: Auf dem ersten Band simuliert $\tilde{M}$ die Berechnung von M, auf dem zweiten die Berechnung von M'. Die 2-Band-Turing-Maschine $\tilde{M}$ führt abwechselnd jeweils einen Simulationsschritt für M und M' aus. Erreicht dabei M oder M' einen Haltezustand, dann hält auch $\tilde{M}$.

Aus dem erreichten Haltezustand kann $\tilde{M}$ schließen, ob die Eingabe w zu L gehört oder nicht. ■

Satz: *Die Sprache $\mathscr{A}_{TM}$ ist rekursiv aufzählbar.*

Beweis:
Jede Universelle Turing-Maschine akzeptiert $\mathscr{A}_{\mathsf{TM}}$. ■

Satz: *Die Klasse der entscheidbaren Sprachen ist eine echte Teilmenge der Klasse der rekursiv aufzählbaren Sprachen.*

Beweis:
$\mathscr{A}_{\mathsf{TM}}$ ist rekursiv aufzählbar, aber nicht entscheidbar. ■

Satz: *Die Sprache $\overline{\mathscr{A}_{TM}} = \{w \mid w \notin \mathscr{A}_{TM}\}$ ist nicht rekursiv aufzählbar.*

Beweis:
Wäre $\overline{\mathscr{A}_{\mathsf{TM}}}$ rekursiv aufzählbar, so wäre $\mathscr{A}_{\mathsf{TM}}$ nach den eben bewiesenen Sätzen entscheidbar, ist es aber nicht. ■

Satz: *Die Klasse der rekursiv aufzählbaren Sprachen ist nicht abgeschlossen unter Komplementbildung.*

Beweis:
$\mathscr{A}_{\mathsf{TM}}$ ist rekursiv aufzählbar, nicht aber ihr Komplement $\overline{\mathscr{A}_{\mathsf{TM}}}$. ■

Abschlusseigenschaften entscheidbarer Sprachen

Satz:
Die Klasse der entscheidbaren Sprachen ist abgeschlossen unter

(a) Vereinigung,
(b) Konkatenation,
(c) Kleene Star,
(d) Komplement und
(e) Schnitt.

Abschlusseigenschaften rekursiv aufzählbarer Sprachen

Satz:
Die Klasse der rekursiv aufzählbaren Sprachen ist abgeschlossen unter

(a) Vereinigung,
(b) Konkatenation,
(c) Kleene Star und
(d) Schnitt.

Unentscheidbare Probleme bei Turing-Maschinen

Satz: *Die Sprache* $\mathscr{A}_{TM,\varepsilon} = \{\langle M\rangle \mid \varepsilon \in L(M)\}$ *ist nicht entscheidbar.*

Beweis:
Wäre $\mathscr{A}_{\mathsf{TM},\varepsilon}$ entscheidbar, so wäre auch $\mathscr{A}_{\mathsf{TM}}$ entscheidbar: Zu gegebenem M und w können wir leicht eine Turingmaschine M_w konstruieren, die, wenn sie mit leerem Band gestartet wird, zunächst w auf das Band schreibt und dann M simuliert.

Dann ist $\langle M,w\rangle \in \mathscr{A}_{\mathsf{TM}}$ genau dann wenn $\langle M_w\rangle \in \mathscr{A}_{\mathsf{TM},\varepsilon}$. ■

Satz: *Die Sprache* $\mathscr{E}_{TM} = \{\langle M\rangle \mid L(M) = \emptyset\}$ *ist nicht entscheidbar.*

Beweis:
Wäre $\mathscr{E}_{\mathsf{TM}}$ entscheidbar, so wäre auch $\mathscr{A}_{\mathsf{TM}}$ entscheidbar: Zu gegebenem M und w können wir leicht eine Turingmaschine M'_w konstruieren, die jede von w verschiedene Eingabe verwirft und bei Eingabe w die Berechnung von M bei Eingabe w simuliert.

Dann ist $\langle M,w\rangle \in \mathscr{A}_{\mathsf{TM}}$ genau dann wenn $\langle M'_w\rangle \notin \mathscr{E}_{\mathsf{TM}}$. ■

Satz: *Die Sprache* $\mathscr{A}_{TM,all} = \{\langle M\rangle \mid L(M) = \Sigma^*\}$ *ist nicht entscheidbar.*

Beweis:
Wäre $\mathscr{A}_{\mathsf{TM},\mathsf{all}}$ entscheidbar, so wäre auch $\mathscr{A}_{\mathsf{TM}}$ entscheidbar: Zu gegebenem M und w können wir leicht eine Turingmaschine M''_w konstruieren, die zunächst jede Eingabe löscht, dann w aufs Band schreibt und die Berechnung von M bei Eingabe w simuliert.

Dann ist $\langle M,w\rangle \in \mathscr{A}_{\mathsf{TM}}$ genau dann wenn $\langle M''_w\rangle \in \mathscr{A}_{\mathsf{TM},\mathsf{all}}$. ■

Satz: *Die Sprache* $\mathscr{Q}_{TM} = \{\langle M_1,M_2\rangle \mid L(M_1) = L(M_2)\}$ *ist nicht entscheidbar.*

Beweis:
Wäre $\mathscr{Q}_{\mathsf{TM}}$ entscheidbar, so wäre auch $\mathscr{A}_{\mathsf{TM},\mathsf{all}}$ entscheidbar: Wir können leicht eine Turingmaschine Y konstruieren, die jede Eingabe akzeptiert.

Dann ist $\langle M\rangle \in \mathscr{A}_{\mathsf{TM},\mathsf{all}}$ genau dann wenn $\langle M,Y\rangle \in \mathscr{Q}_{\mathsf{TM}}$. ■

Reduzierbarkeit

Wir haben die Unentscheidbarkeit der betrachteten Sprachen bei Turing-Maschinen gezeigt, indem wir sie auf die bereits bekannte Unentscheidbarkeit anderer Sprachen *zurückgeführt* haben, insbesondere auf die Unentscheidbarkeit von $\mathscr{A}_{\mathsf{TM}}$.

Um zu zeigen, dass L nicht entscheidbar ist, haben wir gezeigt, dass eine unentscheidbare Sprache U entscheidbar wäre, wenn L entscheidbar wäre.

Definition:
Eine Sprache A heißt abbildungsreduzierbar auf eine Sprache B, falls es eine Turing-berechenbare Funktion $f : \Sigma^* \to \Sigma^*$ gibt, so dass für alle $w \in \Sigma^*$ gilt

$$w \in A \iff f(w) \in B$$

Die Funktion f heißt Reduktion von A auf B.

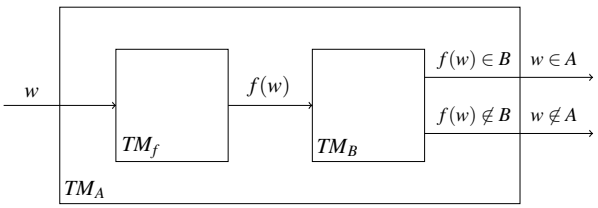
Wir schreiben $A \preceq B$, falls A auf B abbildungsreduzierbar ist.

Satz:
Falls $A \preceq B$ und A ist nicht entscheidbar, dann ist auch B nicht entscheidbar.

Dies folgt unmittelbar aus folgendem

Satz:
Falls $A \preceq B$ und B ist entscheidbar, dann ist auch A entscheidbar.

Beweis:



■

Analog zeigt man

Satz:
Falls $A \preceq B$ und B ist rekursiv aufzählbar, dann ist auch A rekursiv aufzählbar.

Ferner gilt

Lemma: *Die Relation $\preceq$ ist transitiv.*

Aus dem Beweis der Unentscheidbarkeit von $\mathscr{A}_{\mathsf{TM},\varepsilon}$ lässt sich

$$\mathscr{A}_{\mathsf{TM}} \preceq \mathscr{A}_{\mathsf{TM},\varepsilon}$$

ableiten: Wir definieren eine Reduktion $f : \Sigma^* \to \Sigma^*$, die $\langle M, w\rangle$ auf $\langle M_w\rangle$ abbildet und Wörter, die nicht von der Form $\langle M, w\rangle$ sind, auf Wörter, die nicht von der Form $\langle M\rangle$ sind.

Diese Reduktionsfunktion f ist Turing-berechenbar und es gilt für alle $x \in \Sigma^*$

$$x \in \mathscr{A}_{\mathsf{TM}} \iff f(x) \in \mathscr{A}_{\mathsf{TM},\varepsilon}$$

Ferner lassen sich aus den Beweisen zur Unentscheidbarkeit von $\mathscr{E}_{\mathsf{TM}}$, $\mathscr{A}_{\mathsf{TM,all}}$ und $\mathscr{Q}_{\mathsf{TM}}$ Turing-berechenbare Reduktionsfunktionen ableiten, so dass gilt:

$$\begin{aligned} \mathscr{A}_{\mathsf{TM}} &\preceq \overline{\mathscr{E}_{\mathsf{TM}}} \\ \mathscr{A}_{\mathsf{TM}} &\preceq \mathscr{A}_{\mathsf{TM,all}} \\ \mathscr{A}_{\mathsf{TM,all}} &\preceq \mathscr{Q}_{\mathsf{TM}} \end{aligned}$$