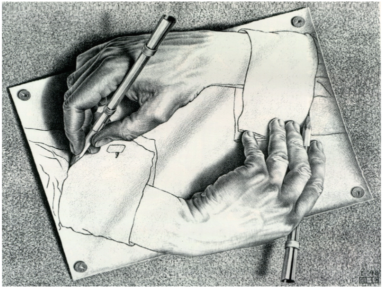
Grundlagen der Theoretischen Informatik

Till Mossakowski

Fakultät für Informatik
Otto-von-Guericke Universität
Magdeburg

Wintersemester 2014/15

Unentscheidbarkeit



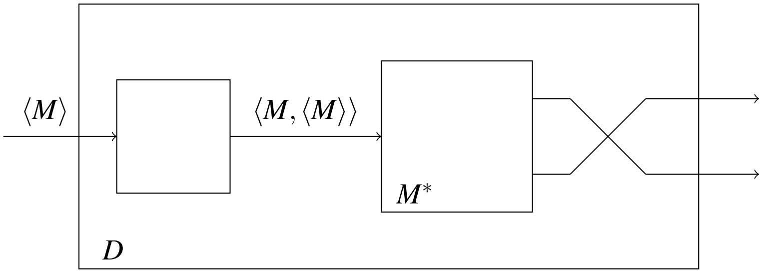
```
#include <stdio.h>
char *s="include <stdio.h>%cchar *s=%c%s%c;%cint main(){printf(s,10,34,s,34,10,10);}%c";
int main(){printf(s,10,34,s,34,10,10);}
```

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

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

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. $\blacksquare$

Unentscheidbare Probleme bei Turing-Maschinen

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

Beweis:
Wäre $\mathscr{A}_{\mathsf{TM,all}}$ 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 $\langle M''_w\rangle \in \mathscr{A}_{\mathsf{TM,all}}$. ■

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

Dann ist $\langle M\rangle \in \mathscr{A}_{\mathsf{TM,all}}$ 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*.*

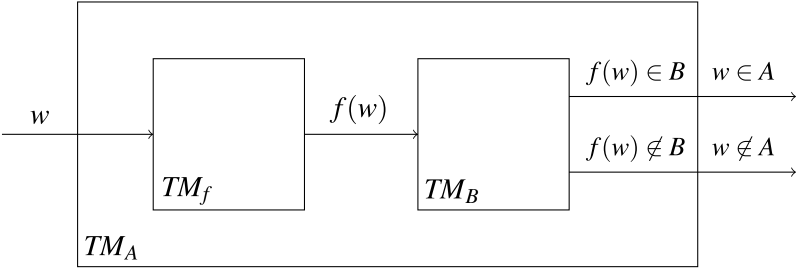
Wir schreiben $A \preceq B$, falls A auf B abbildungsreduzierbar ist.

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:

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

Halten und Akzeptieren

Wenn eine Turing-Maschine ein Wort w nicht akzeptiert, so kann sie w verwerfen oder nie halten. Sei M eine Turing-Maschine, die L akzeptiert. Wir zeigen nun im Wesentlichen, dass wir annehmen dürfen, dass M genau dann hält, wenn das Eingabewort zu L gehört, denn wenn M diese Eigenschaft nicht besitzt, so lässt sich doch aus M leicht eine Turing-Maschine $M^\star$ mit dieser Eigenschaft ableiten, so dass $L = L(M^\star) = L(M)$.

Wir brauchen also in diesem Sinne nicht zwischen Halten und Akzeptieren zu unterscheiden.

Lemma: $\mathscr{A}_{TM} \preceq \mathscr{H}$.

Beweis:
Wir müssen eine Reduktion $f^\star : \Sigma^* \to \Sigma^*$ angeben, so dass für alle $x \in \Sigma^*$ gilt

$$x \in \mathscr{A}_{\mathsf{TM}} \iff f^\star(x) \in \mathscr{H}$$

Falls x nicht von der Form $\langle M, w \rangle$ ist, so ist $f^\star(x) = \varepsilon \notin \mathscr{H}$. Ansonsten ist $f^\star(\langle M, w \rangle) = \langle M^\star, w \rangle$, wobei $M^\star$ genau dann hält, wenn M das Wort w akzeptiert:

$M^\star$ enthält alle Zustände von M und einen weiteren Zustand q_{loop}. $M^\star$ besitzt die gleichen Übergänge wie M, außer dass jeder Übergang, der in M zu q_{reject} führt, nun zu q_{loop} führt. Ferner ist $\delta(q_{\mathsf{loop}}, \sigma) = (q_{\mathsf{loop}}, \sigma, S)$ für alle σ aus dem Bandalphabet. Nun gilt $L(M) = L(M^\star)$. Schließlich ist $f^\star$ Turing-berechenbar. ■

$$\overline{\mathscr{H}} = \{x \mid x \text{ ist von der Form } \langle M, w\rangle \text{ und die Turing-Maschine } M \text{ hält nicht bei Eingabe } w \text{ oder } x \text{ ist nicht von dieser Form}\}$$

Lemma: $\overline{\mathscr{H}}$ *ist nicht rekursiv aufzählbar.*

Beweis:
$\mathscr{H}$ ist rekursiv aufzählbar. Wäre auch $\overline{\mathscr{H}}$ rekursiv aufzählbar, so wäre $\mathscr{H}$ entscheidbar, ist es aber nicht.

$\blacksquare$

Lemma: $\overline{\mathscr{H}} \preceq \mathscr{A}_{TM,all}$.

Beweis:
Wir müssen eine Reduktion $f : \Sigma^* \to \Sigma^*$ angeben, so dass für alle $x \in \Sigma^*$ gilt

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

Falls x nicht von der Form $\langle M, w \rangle$ ist, so ist $f(x) = \langle M_{\mathsf{all}} \rangle$, wobei M_{all} eine Turing-Maschine ist, die bei allen Eingaben hält. Folglich gilt $\langle M_{\mathsf{all}} \rangle \in \mathscr{A}_{\mathsf{TM,all}}$.

Ansonsten ist $f(\langle M, w \rangle) = \langle M_w^{\#} \rangle$, wobei $M_w^{\#}$ eine Turing-Maschine ist, die ihre Eingabe y nach folgendem Verfahren bearbeitet:

1. Kopiere Eingabe y auf ein zweites Band
2. Lösche das Eingabeband und schreibe w auf das Band
3. Simuliere M bei Eingabe w für $|y|$ Schritte oder bis M hält
4. Falls M angehalten hat, gehe in eine Endlosschleife über
5. Ansonsten halte akzeptierend an

M hält bei Eingabe w nicht, genau dann wenn $M_w^\#$ alle Eingaben akzeptiert. Ferner ist die Reduktion f, die $\langle M, w\rangle$ auf $\langle M_w^\# \rangle$ abbildet, Turing-berechenbar. ■

Lemma: *$\mathscr{A}_{\mathit{TM,all}}$ ist nicht rekursiv aufzählbar.*

Sei

$$\begin{aligned}\mathscr{H}_\varepsilon &= \{\langle M\rangle \mid M \text{ hält bei Eingabe } \varepsilon\}\\ \mathscr{H}_{\mathsf{all}} &= \{\langle M\rangle \mid M \text{ hält bei allen Eingaben}\}\end{aligned}$$

Dann gilt

$$\begin{array}{lclcrcl}\mathscr{H} & \preceq & \mathscr{H}_\varepsilon & \qquad & \mathscr{A}_{\mathsf{TM},\varepsilon} & \preceq & \mathscr{H}_\varepsilon\\ \mathscr{H} & \preceq & \mathscr{H}_{\mathsf{all}} & \qquad & \mathscr{A}_{\mathsf{TM,all}} & \preceq & \mathscr{H}_{\mathsf{all}}\end{array}$$

und somit

Lemma: $\mathscr{H}_\varepsilon$ *ist nicht entscheidbar.*

Lemma: $\mathscr{H}_{all}$ *ist nicht rekursiv aufzählbar.*

Satz von Rice

Satz: [Rice]
Sei $\mathcal{T}$ eine echte, nicht-leere Teilmenge der Klasse aller rekursiv aufzählbaren Sprachen. Dann ist

$$\mathscr{L}_{\mathcal{T}} = \{\langle M\rangle \mid L(M) \in \mathcal{T}\}$$

nicht entscheidbar.

Beweis:
Wäre $\mathscr{L}_{\mathcal{T}}$ entscheidbar, so wäre auch $\mathscr{A}_{\mathsf{TM}}$ entscheidbar:

O.B.d.A. sei $\emptyset \notin \mathcal{T}$. Sei L eine Sprache in $\mathcal{T}$ und sei M_L eine Turing-Maschine, die L akzeptiert.

Zu gegebenem M und w können wir eine Turingmaschine $A_{M,w}$ konstruieren, die zunächst ihre Eingabe x auf einem zweiten Band sichert, dann die Berechnung von M bei Eingabe w simuliert, und, falls diese Berechnung in einem akzeptierenden Haltezustand endet, anschließend die Berechnung von M_L bei Eingabe x ausführt.

$L(A_{M,w})$ ist entweder leer, also nicht in $\mathcal{T}$, oder L, also in $\mathcal{T}$, je nachdem ob M bei Eingabe w akzeptierend hält oder nicht.

Also ist $\langle M, w\rangle \in \mathscr{A}_{\mathsf{TM}}$ genau dann wenn $\langle A_{M,w}\rangle \in \mathscr{L}_{\mathcal{T}}$. $\blacksquare$

Typischerweise ist $\mathcal{T}$ durch eine Eigenschaft definiert. Um den Satz von Rice anwenden zu können, muss nachgewiesen werden, dass mindestens eine, aber nicht alle von Turing-Maschinen akzeptierten Sprachen diese Eigenschaft besitzen.

Beispiel:

Satz: *Die Sprache* $\{\langle M\rangle \mid L(M)$ *ist regulär*$\}$ *ist nicht entscheidbar.*

Beweis:
Die Behauptung folgt aus dem Satz von Rice mit $\mathcal{T} = REG$: Die Sprachklasse REG ist weder leer, denn $\mathcal{L}(a^*)$ liegt in REG, noch umfasst REG alle rekursiv aufzählbaren Sprachen, denn die Sprache $\{a^n b^n \mid n \geq 0\}$ ist rekursiv aufzählbar, liegt aber nicht in REG. Also ist der Satz von Rice anwendbar. ■

Unentscheidbare Probleme bei allgemeinen Grammatiken

Da wir zu einer Turing-Maschine M eine allgemeine Grammatik G konstruieren können, so dass $L(G) = L(M)$, lassen sich die Unentscheidbarkeitsresultate von Turing-Maschinen auf allgemeine Grammatiken übertragen:

$$\mathscr{A}_{\mathsf{TM}} \;\preceq\; \{\langle G, w\rangle \mid G \text{ ist eine Grammatik und } w \in L(G)\}$$

Satz:
Die Sprache $\mathscr{A}_{UG} = \{\langle G, w\rangle \mid G$ *ist eine Grammatik und* $w \in L(G)\}$ *ist nicht entscheidbar.*

Ebenso gilt

Satz: *Die Sprache* $\{\langle G_1, G_2\rangle \mid G_1$ *und* G_2 *sind Grammatiken und* $L(G_1) = L(G_2)\}$ *ist nicht entscheidbar.*

Aus dem Satz von Rice folgt ferner beispielsweise

Satz: *Die Sprachen*

$\{\langle G\rangle \mid G$ *ist eine Grammatik und* $\varepsilon \in L(G)\}$,
$\{\langle G\rangle \mid G$ *ist eine Grammatik und* $L(G) = \emptyset\}$,
$\{\langle G\rangle \mid G$ *ist eine Grammatik und* $L(G)$ *ist endlich*$\}$,
$\{\langle G\rangle \mid G$ *ist eine Grammatik und* $L(G)$ *ist regulär*$\}$,
$\{\langle G\rangle \mid G$ *ist eine Grammatik und* $L(G)$ *ist kontextfrei*$\}$ *und*
$\{\langle G\rangle \mid G$ *ist eine Grammatik und* $L(G)$ *ist entscheidbar*$\}$

sind jeweils nicht entscheidbar.

Postsches Korrespondenzproblem

Definition:
Sei Σ ein Alphabet. Eine nicht-leere Folge von geordneten Wortpaaren $((x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n))$ mit $x_i,y_i \in \Sigma^+$ heißt Postsches Korrespondenzsystem.

Eine nicht-leere Folge $(i_1,i_2,\ldots,i_k)$ von Indizes aus $\{1,2,\ldots,n\}$ heißt Lösung des Korrespondenzsystems, falls

$$x_{i_1}x_{i_2}\ldots x_{i_k} = y_{i_1}y_{i_2}\ldots y_{i_k}$$

Eine Lösung $(i_1,i_2,\ldots,i_k)$ des Korrespondenzsystems heißt speziell, falls $i_1 = 1$.

Beispiele:

Das Korrespondenzsystem $((11,111), (100,001), (111,11))$ besitzt eine spezielle Lösung, nämlich $(1,2,3)$:

$$\begin{array}{c} 11100111 \\ 11100111 \end{array}$$

Das Korrespondenzsystem $((00,0), (001,11), (1000,011))$ besitzt keine Lösung!

Das Korrespondenzsystem $((011,1), (0,00), (0,11), (11,0))$ besitzt eine Lösung, nämlich $(2,1,1,4,3,3,2,2)$, aber keine spezielle.

$$\begin{array}{c} 0011011110000 \\ 0011011110000 \end{array}$$

Beim *Postschen Korrespondenzproblem* ist zu entscheiden, ob ein gegebenes Postsches Korrespondenzsystem eine Lösung besitzt.

Beim *modifizierten Postschen Korrespondenzproblem* wird für ein gegebenes Postsches Korrespondenzsystem nach der Existenz einer speziellen Lösung gefragt, also nach einer Folge $(1, i_2, \ldots, i_k)$ von Indizes gesucht, so dass $x_1 x_{i_2} \ldots x_{i_k} = y_1 y_{i_2} \ldots y_{i_k}$.

Im Folgenden sei $\langle P \rangle$ eine Kodierung des Postschen Korrespondenzsystem P, z.B. über dem Alphabet $\{0,1\}$.

$$\begin{aligned} \text{PCP} &= \{\langle P \rangle \mid P \text{ besitzt eine Lösung}\} \\ \text{MPCP} &= \{\langle P \rangle \mid P \text{ besitzt eine spezielle Lösung}\} \end{aligned}$$

Satz: MPCP *ist unentscheidbar.*

Beweis:
Wir zeigen $\mathscr{A}_{\mathsf{UG}} = \{\langle G, w \rangle \mid w \in L(G)\} \preceq \text{MPCP}$.

Wir müssen eine geeignete Reduktionsfunktion $f : \Sigma^* \to \Sigma^*$ angeben. Falls $x \in \Sigma^*$ keine Grammatik kodiert, ist $f(x)$ keine Kodierung eines Postschen Korrespondenzsystem.

Ansonsten sei $G = (V, \Gamma, R, S)$ die in x kodierte Grammatik und w das in x kodierte Wort. Dann ist $f(x)$ die Kodierung des folgenden Postschen Korrespondenzsystem P:

O.B.d.A. nehmen wir an, dass $\vdash$ und $\dashv$ nicht in Γ enthalten sind. Das erste Wortpaar in P ist

$$(\vdash, \vdash S \Rightarrow)$$

Ferner enthält P das Wortpaar $(a,\, a)$ für alle $a \in \Gamma$, das Wortpaar $(A,\, A)$ für alle $A \in V$, sowie das Wortpaar $(\Rightarrow,\, \Rightarrow)$.

Für alle $u \to_G v$ in R enthält P das Wortpaar

$$(u,\, v)$$

Schließlich enthält G das Wortpaar

$$(\Rightarrow w \dashv, \dashv)$$

P besitzt eine spezielle Lösung genau dann wenn $w \in L(G)$. Die Funktion f ist Turing-berechenbar und es gilt für alle $x \in \Sigma^*$

$$x \in \mathscr{A}_{\mathsf{UG}} \iff f(x) \in \mathrm{MPCP}$$

Mit einem Algorithmus, der das modifizierte Postsche Korrespondenzproblem löst, könnte man also entscheiden, ob ein Wort w zu der von einer Grammatik G erzeugten Sprache $L(G)$ gehört. ■

Beispiel:

$$G = (\{S,B,C\},\{a,b,c\},R,S)$$

$$\begin{array}{rcl} S & \to & \varepsilon \\ S & \to & aSBC \\ CB & \to & BC \\ aB & \to & ab \\ bB & \to & bb \\ bC & \to & bc \\ cC & \to & cc \end{array}$$

$$w = aabbcc$$

$(\vdash, \vdash S \Rightarrow)$
$(a,\ a)$
$(b,\ b)$
$(c,\ c)$
$(S,\ S)$
$(B,\ B)$
$(C,\ C)$
$(S,\ \varepsilon)$
$(S,\ aSBC)$
$(CB,\ BC)$
$(aB,\ ab)$
$(bB,\ bb)$
$(bC,\ bc)$
$(cC,\ cc)$
$(\Rightarrow,\ \Rightarrow)$
$(\Rightarrow aabbcc \dashv,\ \dashv)$

$$\vdash S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaBCBC \Rightarrow aaBBCC \Rightarrow$$

$$\vdash S \Rightarrow aSBC \Rightarrow aaSBCBC \Rightarrow aaBCBC \Rightarrow aaBBCC \Rightarrow aabBCC \Rightarrow$$

$$aabBCC \Rightarrow aabbCC \Rightarrow aabbcC \Rightarrow aabbcc \dashv$$

$$aabbCC \Rightarrow aabbcC \Rightarrow aabbcc \dashv$$

Lemma: $\text{MPCP} \preceq \text{PCP}$.

Beweis:
O.B.d.A. nehmen wir an, dass # und $ nicht in den Wortpaaren des Postschen Korrepondenzsystems P vorkommen.

Zu $P = ((u_1, v_1), \ldots, (u_n, v_n))$ konstruieren wir ein Postsches Korrepondenzsystems P', so dass P genau dann eine spezielle Lösung besitzt, wenn P' eine Lösung besitzt.

Für $w = \sigma_1\sigma_2\cdots\sigma_m \in \Gamma^+$ mit $\sigma_i \in \Gamma$ für $i = 1, \ldots, m$ sei

$$g_L^{\#}(w) = \#\sigma_1\#\sigma_2\ldots\#\sigma_m$$

und

$$g_R^{\#}(w) = \sigma_1\#\sigma_2\#\ldots\sigma_m\#$$

$P' = ((\#g_R^{\#}(u_1),\ g_L^{\#}(v_1)),\ (g_R^{\#}(u_1),\ g_L^{\#}(v_1)),\ (g_R^{\#}(u_2),\ g_L^{\#}(v_2)),\ldots$
$\ldots,\ (g_R^{\#}(u_n),\ g_L^{\#}(v_n)),\ (\$,\#\$))$.

Falls $(1, i_2, \ldots, i_k)$ eine spezielle Lösung für P ist, so ist $(1, i_2+1, \ldots, i_k+1, n+2)$ eine Lösung für P'.

Falls $(j_1, j_2, \ldots, j_k)$ eine Lösung für P' ist, so muss $j_1 = 1$ und $j_k = n+2$ gelten. Dann ist $(1, j_2-1, \ldots, j_{k-1}-1)$ eine spezielle Lösung für P.

Die Reduktionsfunktion f bildet $\langle P \rangle$ auf $\langle P' \rangle$ ab. $\blacksquare$

Beispiel:

$P =$
$((1, 101), (10, 00), (011, 11)).$

$P' =$
$((\#1\#, \#1\#0\#1), (1\#, \#1\#0\#1), (1\#0\#, \#0\#0), (0\#1\#1\#, \#1\#1), (\$, \#\$)).$

$$\begin{array}{c}\#1\#0\#1\#1\#1\#0\#0\#1\#1\#\$ \\ \#1\#0\#1\#1\#1\#0\#0\#1\#1\#\$\end{array}$$

Aus dem Gezeigten ergibt sich

Satz: PCP *ist nicht entscheidbar.*

Ferner gilt

Satz: PCP *ist rekursiv aufzählbar.*

Unentscheidbare Probleme bei kontextfreien Grammatiken

Satz:
Die Sprache $\{\langle G_1, G_2\rangle \mid G_1$ *und* G_2 *sind kontextfreie Grammatiken und* $L(G_1) \cap L(G_2) = \emptyset\}$ *ist nicht entscheidbar.*

Beweis:

Wir zeigen

$$\text{PCP} \preceq \{\langle G_1, G_2\rangle \mid L(G_1) \cap L(G_2) \neq \emptyset\}$$

Sei $P = ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ ein Postsches Korrespondenzproblem über dem Alphabet Σ.

Zu P konstruieren wir Grammatiken

$$G_1 = (\{S_1, A, B\}, \Sigma \cup \{a_1, \ldots, a_n, \$\}, R_1, S_1)$$

und

$$G_2 = (\{S_2, T\}, \Sigma \cup \{a_1, \ldots, a_n, \$\}, R_2, S_2)$$

R_1 umfasst die Regeln

$$\begin{array}{rcl} S_1 & \rightarrow & A\$B \\ A & \rightarrow & a_1Ax_1 \mid \cdots \mid a_nAx_n \mid a_1x_1 \mid \cdots \mid a_nx_n \\ B & \rightarrow & y_1^RBa_1 \mid \cdots \mid y_n^RBa_n \mid y_1^Ra_1 \mid \cdots \mid y_n^Ra_n \end{array}$$

R_2 umfasst die Regeln

$$\begin{array}{rcl} S_2 & \rightarrow & a_1S_2a_1 \mid \cdots \mid a_nS_2a_n \mid T \\ T & \rightarrow & \sigma T\sigma \text{ für alle } \sigma \in \Sigma \\ T & \rightarrow & \$ \end{array}$$

$$L(G_1) = \{a_{i_k} \dots a_{i_1} x_{i_1} \dots x_{i_k} \$ y_{j_m}^R \dots y_{j_1}^R a_{j_1} \dots a_{j_m} \mid k,m \geq 1, i_\mu, j_\nu \in \{1,\dots,n\}\ \}$$

$$L(G_2) = \{uv\$v^R u^R \mid u \in \{a_1,\dots,a_n\}^*,\ v \in \Sigma^*\}$$

P besitzt die Lösung $(i_1, i_2, \dots, i_k)$ genau dann wenn $L(G_1) \cap L(G_2)$ das Wort $a_{i_k} \dots a_{i_1} x_{i_1} \dots x_{i_k} \$ y_{i_k}^R \dots y_{i_1}^R a_{i_1} \dots a_{i_k}$ enthält.

Da alle Wörter in $L(G_1) \cap L(G_2)$ von obiger Form sind, ist $L(G_1) \cap L(G_2) \neq \emptyset$ genau dann wenn P eine Lösung besitzt.

Die zu dieser Transformation gehörige Reduktionsfunktion ist Turing-berechenbar. Also ist $\{\langle G_1, G_2 \rangle \mid L(G_1) \cap L(G_2) \neq \emptyset\}$ nicht entscheidbar, also ist auch $\{\langle G_1, G_2 \rangle \mid L(G_1) \cap L(G_2) = \emptyset\}$ nicht entscheidbar. ■

Satz:
Die Sprache $\mathscr{A}_{\mathit{CFG,all}} = \{\langle G\rangle \mid G$ ist eine kontextfreie Grammatik über dem Alphabet Σ mit $L(G) = \Sigma^*\}$ ist nicht entscheidbar.

Beweisskizze:

Wir zeigen, dass $\mathscr{E}_{\mathsf{TM}} = \{\langle M\rangle \mid L(M) = \emptyset\}$ entscheidbar wäre, wenn $\mathscr{A}_{\mathsf{CFG,all}}$ entscheidbar wäre.

Ein Wort w wird von einer Turingmaschine M akzeptiert, genau dann wenn es eine akzeptierende Berechnung für w gibt, also eine Folge $C_0, C_1, \ldots, C_n$ von Konfigurationen mit

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \cdots \vdash_M C_n$$

so dass C_0 die Startkonfiguration für w und C_n eine akzeptierende Haltekonfiguration ist.

Wir kodieren die Konfigurationen von M in naheliegender Weise als Kodierung des relevanten Bandinhalts, in den der aktuelle Zustand als Markierung der Position des Schreib-Lesekopfs integriert ist. $\langle C \rangle$ bezeichne die Kodierung der Konfiguration C.

Eine Berechnung $C_0 \vdash_M C_1 \vdash_M \cdots \vdash_M C_n$ gerader Länge kodieren wir als Wort

$$\langle C_0 \rangle \# \langle C_1 \rangle^R \# \langle C_2 \rangle \# \langle C_3 \rangle^R \# \langle C_4 \rangle \# \cdots \# \langle C_n \rangle$$

und eine Berechnung $C_0 \vdash_M C_1 \vdash_M \cdots \vdash_M C_n$ ungerader Länge als

$$\langle C_0 \rangle \# \langle C_1 \rangle^R \# \langle C_2 \rangle \# \langle C_3 \rangle^R \# \langle C_4 \rangle \# \cdots \# \langle C_n \rangle^R$$

Die Sprache $L_{\overline{M}}$, die alle Wörter über dem Kodierungsalphabet umfasst, die keiner gültigen Berechnung von M entsprechen, ist kontextfrei, denn es gibt einen Kellerautomaten, der genau diese Wörter akzeptiert.

Ein Wort w gehört zu $L_{\overline{M}}$, falls

- w nicht von der gesuchten Form ist, oder
- C_0 keine Startkonfiguration ist, oder
- C_n keine akzeptierende Haltekonfiguration ist, oder
- es ein $i \in [0..n-1]$ gibt, so dass $C_i \vdash_M C_{i+1}$ nicht gilt.

Jede dieser Bedingungen lässt sich mit einem Kellerautomaten überprüfen.

Also gibt es einen Kellerautomaten, der $L_{\overline{M}}$ akzeptiert.
Zu diesem Kellerautomaten können wir eine kontextfreie Grammatik G mit $L(G) = L_{\overline{M}}$ konstruieren.

$L_{\overline{M}}$ enthält alle Wörter über dem Kodierungsalphabet genau dann wenn $L(M) = \emptyset$. Zu gegebenem $\langle M \rangle$ können wir wie beschrieben eine kontextfreie Grammatik G konstruieren, so dass

$$\langle G \rangle \in \mathscr{A}_{\mathsf{CFG,all}} \quad \text{g.d.w.} \quad \langle M \rangle \in \mathscr{E}_{\mathsf{TM}}$$

Wäre $\mathscr{A}_{\mathsf{CFG,all}}$ entscheidbar, so wäre also auch $\mathscr{E}_{\mathsf{TM}}$ entscheidbar. $\blacksquare$

Ferner kann man zeigen:

Satz: *Die Sprachen*

$\{\langle G_1, G_2\rangle \mid G_1$ *und* G_2 *sind kontextfreie Grammatiken und*
$|L(G_1) \cap L(G_2)| = \infty\}$,

$\{\langle G_1, G_2\rangle \mid G_1$ *und* G_2 *sind kontextfreie Grammatiken und*
$L(G_1) \cap L(G_2)$ *ist kontextfrei*$\}$,

$\{\langle G_1, G_2\rangle \mid G_1$ *und* G_2 *sind kontextfreie Grammatiken und*
$L(G_1) \subseteq L(G_2)\}$,

$\{\langle G_1, G_2\rangle \mid G_1$ *und* G_2 *sind kontextfreie Grammatiken und*
$L(G_1) = L(G_2)\}$,

$\{\langle G\rangle \mid G$ *ist eine kontextfreie Grammatik und* G *ist mehrdeutig*$\}$,

$\{\langle G\rangle \mid G$ *ist eine kontextfreie Grammatik und* $L(G)$ *ist regulär*$\}$ *und*

$\{\langle G\rangle \mid G$ *ist eine kontextfreie Grammatik und* $\overline{L(G)}$ *ist kontextfrei*$\}$

sind jeweils nicht entscheidbar.