Post'sches Korrespondenzproblem

Hans U. Simon (RUB)

mit Modifikationen von

Maike Buchin (RUB)

Lehrstuhl Mathematik und Informatik

Homepage: http://www.ruhr–uni–bochum.de/lmi

PKP und MPKP

Postsches Korrespondenzproblem (PKP)

Entscheide zu einer gegebenen Folge

$$K = [(x_1, y_1), \ldots, (x_k, y_k)]$$

von Wortpaaren über einem endlichen Alphabet Σ, ob es eine Folge

$$i_1, \ldots, i_n \in [1 : k]$$

von Indizes, genannt „Lösung“, gibt, so dass

$$x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n} \ .$$

Modifiziertes Postsches Korrespondenzproblem (MPKP)

Wie PKP, außer dass die Indexfolge mit $i_1 = 1$ beginnen muss.

Beispiel 1

Zu

$$K = [(1, 111), (10111, 10), (10, 0)]$$

ist $(2, 1, 1, 3)$ eine passende Indexfolge:

$$\overbrace{10111}^{x_2}\overbrace{1}^{x_1}\overbrace{1}^{x_1}\overbrace{10}^{x_3} = 101111110 = \overbrace{10}^{y_2}\overbrace{111}^{y_1}\overbrace{111}^{y_1}\overbrace{0}^{y_3}$$

Beispiel 2

Zu

$$K = [(10, 101), (011, 11), (101, 011)]$$

gibt es keine passende Indexfolge (Zugzwangargument):

1. Jede potenzielle Lösung müßte beginnen mit $i_1 = 1$:

$$x_1 = 10, \; y_1 = 101$$

2. Wann immer die y-Sequenz eine 1 Vorsprung hat, ist die einzig aussichtsreiche Fortsetzung

$$\begin{array}{lll} x\text{-Sequenz}: & \ldots & \overbrace{101}^{x_3} \\ y\text{-Sequenz}: & \ldots & 1\underbrace{011}_{y_3} \end{array},$$

was den Vorsprung von der y-Sequenz auf ewig reproduziert.

Beispiel 3

Zu

$$K = [(001, 0), (01, 011), (01, 101), (10, 001)]$$

gibt es eine passende Indexfolge $i_1, \ldots, i_n$, aber erst ab $n = 66$.

Wer findet die Lösung ?

Hauptresultat

Wir werden nachweisen die Reduktionskette:

$$H \leq MPKP \leq PKP$$

Folgerung: MPKP und PKP sind unentscheidbar.

Bemerkungen:

1. Die Unentscheidbarkeit ergibt sich bereits für binäres Alphabet (wie sich zeigen wird).
2. Bei unärem Alphabet hingegen sind MPKP und PKP entscheidbar. (Der Beweis hierfür wäre eine gute Übungsaufgabe.)
3. Für die Semi-Entscheidbarkeit von PKP können immer länger werdende Indexfolgen systematisch getestet werden.

Reduktion von U auf MPKP

Ziel: Entwurf einer Reduktionsabbildung f, die Eingaben von U der Form $w\#x$, so auf Eingaben von MPKP abbildet, dass gilt:

$$x \in T(M_w) \Leftrightarrow f(w\#x) \text{ hat eine Lösung } . \tag{1}$$

Notation: Im Folgenden schreiben wir einfach M statt M_w und benutzen die üblichen Symbole für die Komponenten von M.

Idee: Um (1) zu erzwingen, werden die x- und y-Sequenzen Konfigurationsfolgen von M entsprechen, wobei die y-Sequenz immer eine Konfiguration Vorsprung hat. Der x-Sequenz erlauben wir erst nach Akzeptanz von M diesen Vorsprung einzuholen.

Normierung der Turing-Maschine M

Indem wir (falls nötig) die TM M leicht normieren (ohne ihr Akzeptanzverhalten auf Eingabe x zu verändern), können wir o.E. voraussetzen:

1. M bewegt in jedem Schritt den Kopf.
2. M druckt niemals ihr Leerzeichen $\square$.

Diese Normierung vereinfacht die folgende Konstruktion der MPKP-Eingabeinstanz $f(w\#x)$.

Die MPKP-Eingabeinstanz K=f(w#x)

Die Stringpaare von K zerfallen in fünf Gruppen:

- das Anfangspaar $(\#, \#z_0x\#)$
- Kopierpaare (X, X) für alle $X \in \Gamma \cup \{\#\}$
- Überführungspaare

$$\begin{aligned}
&(zY, Y'z') && \text{, falls } \delta(z, Y) = (z', Y', R)\\
&(XzY, z'XY') && \text{, falls } \delta(z, Y) = (z', Y', L)\\
&(z\#, Y'z'\#) && \text{, falls } \delta(z, \square) = (z', Y', R)\\
&(Xz\#, z'XY'\#) && \text{, falls } \delta(z, \square) = (z', Y', L)
\end{aligned}$$

  für alle $z \in Z \setminus S$, $z' \in Z$, $X, Y, Y' \in \Gamma \setminus \{\square\}$.

- Löschpaare (XsY, s), $(Xs\#, s\#)$, $(\#sY, \#s)$
  für alle $s \in S$, $X, Y \in \Gamma \setminus \{\square\}$.
- Abschlusspaare $(s\#\#, \#)$ für alle $s \in S$.

Intuition hinter diesem Design

- Das Anfangspaar dient dazu, der y-Sequenz eine Konfiguration (hier die Anfangskonfiguration) Vorsprung zu geben.
- Die Kopier- und Überführungspaare dienen dazu, beide Sequenzen um eine Konfiguration zu verlängern.
- Die Lösch- und Abschlußpaare sollen die x-Sequenz den Vorsprung aufholen lassen, sofern einen Zustand aus S erreicht wurde.

Um zu verifizieren, dass dieser Plan aufgeht, benötigen wir das folgende Konzept der *partiellen Lösung* für K.

Partielle Lösung für K=f(w#x)

Stringpaar $(x, y) \in \Sigma^* \times \Sigma^*$ heißt eine *partielle Lösung* für K ist, wenn gilt:

1. x ist Anfangswort von y.
2. Es existiert ein $n \geq 1$ und $i_1, \ldots, i_n$ mit $i_1 = 1$ (das Anfangspaar), so dass x die x-Sequenz und y die y-Sequenz zu $i_1, \ldots, i_n$ ist.

Zentrale Beobachtung:
Falls M aus Startkonfiguration z_0x die Folgekonfigurationen

$$\alpha_1 z_1 \beta_1, \alpha_2 z_2 \beta_2, \ldots, \alpha_k z_k \beta_k \text{ mit } z_0, \ldots, z_{k-1} \notin S$$

produziert, dann besitzt K eine partielle Lösung der Form

$$\begin{aligned}(x, y) \quad = \quad & (\#z_0 w \# \alpha_1 z_1 \beta_1 \# \ldots \# \alpha_{k-1} z_{k-1} \beta_{k-1} \# \ , \\ & \#z_0 w \# \alpha_1 z_1 \beta_1 \# \ldots \# \alpha_{k-1} z_{k-1} \beta_{k-1} \# \alpha_k z_k \beta_k \#)\end{aligned} \tag{2}$$

Induktiver Beweis der zentralen Beobachtung

Der Beweis erfolgt durch Induktion nach k.

k = 0: $(x, y) = (\#, \#z_0 w\#)$ realisiert durch das Anfangspaar.

Schritt von k auf k+1: Sei per Induktionsvoraussetzung eine partielle Lösung der Form (2) gegeben und $z_k \notin S$. Wir können die x-Sequenz um $\alpha_k z_k \beta_k \#$ und die y-Sequenz um $\alpha_{k+1} z_{k+1} \beta_{k+1} \#$ verlängern, indem wir

- die identischen Teile von Konfigurationen k und $k+1$ mit den Kopierpaaren aufbauen,
- die verschiedenen Teile (lokale Umgebung der Zustandssymbole z_k, z_{k+1}, weil dort jeweils der Kopf von M positioniert ist) mit dem eindeutig bestimmten Überführungspaar aufbauen.

Auf diese Weise erhalten wir die partielle Lösung

$$(x', y') = (y, y\alpha_{k+1} z_{k+1} \beta_{k+1} \#) \ .$$

Damit ist der induktive Beweis abgeschlossen.

Illustration des Induktionsschrittes

Es sei $ABCzDEF\#$ die Konfiguration, welche den Vorsprung der y-Sequenz ausmacht. Wir nehmen an, dass die Turing-Tafel von M die Aktion

$$\delta(z, D) = (z', D', L) \tag{3}$$

vorschreibt. Wir machen drei „Schnappschüsse" der x- und y-Sequenz:

x-Sequenz (Einsatz Kopierpaare):	...	AB
y-Sequenz (Einsatz Kopierpaare):	...	$ABCzDEF\#AB$
x-Sequenz (Einsatz Überführungspaar):	...	$ABCzD$
y-Sequenz (Einsatz Überführungspaar:	...	$ABCzDEF\#ABz'CD'$
x-Sequenz (Einsatz Kopierpaare):	...	$ABCzDEF\#$
y-Sequenz (Einsatz Kopierpaare):	...	$ABCzDEF\#ABz'CD'EF\#$

Nachweis der Reduktionseigenschaften von f

Behauptung 1 Falls $x \in T(M)$, dann besitzt K eine Lösung.

Falls $x \in T(M)$, erhalten wir irgendwann eine partielle Lösung der Form

$$(y, y\alpha s\beta\#) \text{ mit } s \in E, \alpha, \beta \in \Gamma^* \ .$$

Nun können wir die Kopierpaare und die Löschpaare einsetzen, um den Vorsprung $\alpha s\beta\#$ zu vermindern:

- Jede Anwendung eines Löschpaares vermindert den Vorsprung um ein α- oder β-Symbol.
- Irgendwann ist der Vorsprung auf $s\#$ zusammengeschmolzen und die Sequenzen haben die Form

  $$(y', y's\#) \ .$$

  Anwendung des Abschlußpaares für s egalisiert die Sequenzen:

  $$(y's\#\#, y's\#\#)$$

Illustration der „Aufholjagd“

x-Sequenz:	...	
y-Sequenz (Endkonfiguration als Vorsprung):	...	$ABsD\#$
x-Sequenz (Einsatz Kopierpaar):	...	A
y-Sequenz (Einsatz Kopierpaar):	...	$ABsD\#A$
x-Sequenz (Einsatz Löschpaar):	...	$ABsD$
y-Sequenz (Einsatz Löschpaar):	...	$ABsD\#As$
x-Sequenz (Einsatz Kopierpaar):	...	$ABsD\#$
y-Sequenz (Einsatz Kopierpaar):	...	$ABsD\#As\#$
x-Sequenz (Einsatz Löschpaar):	...	$ABsD\#As\#$
x-Sequenz (Einsatz Löschpaar):	...	$ABsD\#As\#s\#$
x-Sequenz (Einsatz Abschlusspaar):	...	$ABsD\#As\#s\#\#$
x-Sequenz (Einsatz Abschlusspaar):	...	$ABsD\#As\#s\#\#$

Nachweis der Reduktionseigenschaften von f (fortgesetzt)

Behauptung 2 Falls K eine Lösung besitzt, dann gilt $x \in T(M)$.

Indirekter Beweis: Wir zeigen, dass K keine Lösung besitzt, falls $x \notin T(M)$.

Da wir bei der Produktion von partiellen Lösungen keine Freiheiten hatten (Anwendung von anderen Paaren als die beschriebenen führt immer direkt in eine Sackgasse), besteht der einzige Lösungsversuch im Produzieren partieller Lösungen der Form (2). Solange dabei kein Zustand aus E erreicht wird, kommen die Löschpaare nicht zum Einsatz und die y-Sequenz hat daher stets eine Konfiguration Vorsprung. Eine Egalisierung der Sequenzen kann also nicht stattfinden.

Reduktion von MPKP auf PKP

Die Eingabeinstanz von MPKP über Alphabet Σ sei

$$K = [(x_1, y_1), \ldots, (x_k, y_k)] \ .$$

Seien $\#, \$ \notin \Sigma$ zwei neue Symbole. Dann soll

$$f(K) = [(x'_0, y'_0), (x'_1, y'_1), \ldots, (x'_k, y'_k), (x'_{k+1}, y'_{k+1})]$$

die folgende Eingabe von PKP sein:

- Für $i = 1, \ldots, k$ entsteht x'_i, indem **hinter** jedem Buchstaben von x_i Symbol # eingefügt wird; y'_i entsteht aus y_i, indem **vor** jedem Buchstaben von y_i Symbol # eingefügt wird.
- $x'_0 = \#x'_1,\ x'_{k+1} = \$,\ y'_0 = y'_1,\ y'_{k+1} = \#\$.$

Es folgt ein Beispiel.

x_1	$=$	10111	x'_1	$=$	$1\#0\#1\#1\#1\#$	x'_0	$=$	$\#1\#0\#1\#1\#1\#$
y_1	$=$	10	y'_1	$=$	$\#1\#0$	y'_0	$=$	$\#1\#0$
x_2	$=$	1	x'_2	$=$	$1\#$			
y_2	$=$	111	y'_2	$=$	$\#1\#1\#1$			
x_3	$=$	10	x'_3	$=$	$1\#0\#$	x'_4	$=$	$\$$
y_3	$=$	0	y'_3	$=$	$\#0$	y'_4	$=$	$\#\$$

$(1, 2, 2, 3)$ ist eine passende Indexfolge für $K = [(x_1, y_1), (x_2, y_2), (x_3, y_3)]$:

$$\overbrace{10111}^{x_1}\overbrace{1}^{x_2}\overbrace{1}^{x_2}\overbrace{10}^{x_3} = 101111110 = \overbrace{10}^{y_1}\overbrace{111}^{y_2}\overbrace{111}^{y_2}\overbrace{0}^{y_3} .$$

$(0, 2, 2, 3, 4)$ ist eine passende Indexfolge für $f(K)$, die folgenden „gepolsterten" Lösungsstring liefert:

$$\overbrace{\#1\#0\#1\#1\#1\#}^{x'_0}\overbrace{1\#}^{x'_2}\overbrace{1\#}^{x'_2}\overbrace{1\#0\#}^{x'_3}\overbrace{\$}^{x'_4} = \overbrace{\#1\#0}^{y'_0}\overbrace{\#1\#1\#1}^{y'_2}\overbrace{\#1\#1\#1}^{y'_2}\overbrace{\#0}^{y'_3}\overbrace{\#\$}^{y'_4}$$

Reduktion von MPKP auf PKP (fortgesetzt)

Aus dem Design von $f(K)$ ergibt sich leicht:

1. Für alle n und alle $i_2, \ldots, i_n \in [1:k]$:

   $$1, i_2, \ldots, i_n \text{ Lösung für } K \Leftrightarrow 0, i_2, \ldots, i_n, k+1 \text{ Lösung für } f(K) \ .$$

2. Jede Lösung (= passende Indexfolge) kürzester Länge für $f(K)$ startet mit Index 0, endet mit Index $k+1$ und verwendet dazwischen nur Indizes aus $\{1, \ldots, k\}$.

Es folgt:
K besitzt eine passende Indexfolge **gdw** $f(K)$ besitzt eine passende Indexfolge.
Da f außerdem berechenbar ist, ergibt sich MPKP$\leq$PKP.

PKP mit binärem Alphabet

01-PKP sei das PKP über dem Alphabet $\Sigma = \{0, 1\}$.

Satz: PKP$\leq$01-PKP.

Beweis: Sei K eine PKP-Eingabeinstanz über einem beliebigen Alphabet der Form $\Sigma = \{a_1, \ldots, a_m\}$. Wähle als $f(K)$ die 01-PKP-Eingabeinstanz, die aus K durch die Substitutionsregel

$$a_j \mapsto 10^j$$

hervorgeht. Die Lösungen (sofern vorhanden) für K und $f(K)$ entsprechen sich (in der offensichtlichen Weise) 1-zu-1. Insbesondere besitzt K eine Lösung **gdw** $f(K)$ eine Lösung besitzt.

Folgerung: 01-PKP ist unentscheidbar.

Unentscheidbare Probleme mit Grammatiken und Automaten

Zu 01-PKP assoziierte kontextfreie Sprachen

Zu einem Wort $w = w_1 \cdots w_n$ bezeichne $\tilde{w} = w_n \cdots w_1$ sein „Spiegelbild".
Beachte: $\tilde{uv} = \tilde{v}\tilde{u}$. Zu der Eingabeinstanz

$$K = [(x_1, y_1), \ldots, (x_k, y_k)]$$

von 01-PKP assoziieren wir die folgenden Sprachen über dem Alphabet $\Sigma = \{0, 1, \$, a_1, \ldots, a_k, \}$:

$$\begin{aligned} L_1[K] &:= \{a_{i_m} \cdots a_{i_1} x_{i_1} \cdots x_{i_m} \$ \tilde{y}_{j_n} \cdots \tilde{y}_{j_1} a_{j_1} \cdots a_{j_n} \mid m, n \geq 1\} \\ L_2[K] &:= \left\{uv\$\tilde{v}\tilde{u} \mid u \in \{a_1, \ldots, a_k\}^+, v \in \{0,1\}^+\right\} \end{aligned}$$

Erinnerung: $i_1, \ldots, i_n$ ist eine Lösung für K **gdw** $x_{i_1} \cdots x_{i_n} = y_{i_1} \cdots y_{i_n}$.

Zentrale Beobachtung: $L_1[K] \cap L_2[K]$ ist identisch zu

$$\{a_{i_m} \cdots a_{i_1} x_{i_1} \cdots x_{i_m} \$ \tilde{y}_{i_m} \cdots \tilde{y}_{i_1} a_{i_1} \cdots a_{i_m} \mid i_1, \ldots, i_n \text{ ist eine Lösung für } K\} .$$

(Deterministische) Kontextfreiheit dieser Sprachen

Wir geben hier kfG's $G_1 = G_1[K]$, $G_2 = G_2[K]$ für $L_1 = L_1[K]$, $L_2 = L_2[K]$ an:

1. G_1 enthält die Regeln

$$S_1 \to A\$B \ , \ A \to a_i A x_i \mid a_i x_i \ , \ B \to \tilde{y}_i B a_i \mid \tilde{y}_i a_i$$

   für $i = 1, \ldots, k$.

2. G_2 enthält die Regeln

$$S_2 \to a_i S_2 a_i \mid a_i R a_i \ , \ R \to 0R0 \mid 1R1 \mid 0\$0 \mid 1\$1$$

   für $i = 1, \ldots, k$.

Bemerkung: Grammatiken G_1 und G_2 sind eindeutig und es lassen sich auch DPDA's $M_1 = M_1[K]$ und $M_2 = M_2[K]$ für L_1 und L_2 angeben. (Der Beweis hierfür wäre eine gute Übungsaufgabe.)

Folgerung: L_1, L_2 sind eindeutige deterministisch kontextfreie Sprachen.

Beispiel

Zu $K = [(1, 111), (10111, 10), (10, 0)]$ erhalten wir bei G_1 die Regeln

$$\begin{aligned} S_1 &\to A\$B \\ A &\to a_1A1|a_2A10111|a_3A10|a_11|a_210111|a_310 \\ B &\to 111Ba_1|01Ba_2|0B|a_3|111a_1|01a_2|0 \end{aligned}$$

und bei G_2 die Regeln

$$\begin{aligned} S_2 &\to a_1S_2a_1|a_2S_2a_2|a_3S_2a_3|a_1Ra_1|a_2Ra_2|a_3Ra_3 \\ R &\to 0R0|1R1|0\$0|1\$1 \ . \end{aligned}$$

Zwei sehr unterschiedliche Szenarios

K hat eine Lösung	K hat keine Lösung
1. $L_1[K] \cap L_2[K] \neq \emptyset$	1'. $L_1[K] \cap L_2[K] = \emptyset$
2. $\lvert L_1[K] \cap L_2[K] \rvert = \infty$	2'. $\lvert L_1[K] \cap L_2[K] \rvert < \infty$
3. $L_1[K] \cap L_2[K]$ ist nicht kontextfrei	3'. $L_1[K] \cap L_2[K]$ ist kontextfrei
4. $\overline{L_1[K]} \cup \overline{L_2[K]}$ ist nicht nicht deterministisch kontextfrei	4'. $\overline{L_1[K]} \cup \overline{L_2[K]}$ ist deterministisch kontextfrei
5. $\overline{L_1[K]} \cup \overline{L_2[K]}$ ist nicht regulär	5'. $\overline{L_1[K]} \cup \overline{L_2[K]}$ ist regulär

Begründungen

- Unsere „zentrale Beobachtung“ liefert 1. und 1'.
- Aus $L_1 \cap L_2 = \emptyset$ (und somit $\overline{L_1} \cup \overline{L_2} = \overline{L_1 \cap L_2} = \Sigma^*$) ergeben sich unmittelbar die Aussagen 2'. bis 5'.
- Zu Aussage 2. nutze aus, dass K unendlich viele Lösungen hat, sofern es mindestens eine Lösung hat.
- Aussage 3. ergibt sich leicht mit dem Pumping–Lemma. (Der Beweis hierfür wäre eine gute Übungsaufgabe.)
- Aussage 4. ergibt sich mit einem Beweis durch Widerspruch:
  Wäre $\overline{L_1} \cup \overline{L_2}$ deterministisch kontextfrei, dann müsste auch $L_1 \cap L_2$ deterministisch kontextfrei sein im Widerspruch zu 3.
- Aussage 5. ergibt sich direkt aus 4.

Folgerungen

Die folgenden Probleme sind unentscheidbar (Reduktion von PKP bzw. Komplement von PKP auf das betreffende Problem):

1. Ist der Durchschnitt zweier (durch DPDAs oder kfGs gegebener) deterministisch kontextfreier Sprachen leer (Schnittproblem für deterministisch kontextfreie Sprachen) ?
2. Ist der Durchschnitt zweier (durch DPDAs oder kfGs gegebener) deterministisch kontextfreier Sprachen endlich ?
3. Ist der Durchschnitt zweier (durch DPDAs oder kfGs gegebener) deterministisch kontextfreier Sprachen kontextfrei (Kontextfreiheit des Durchschnittes) ?
4. Ist eine (durch eine kfG gegebene) kontextfreie Sprache deterministisch kontextfrei ?
5. Ist eine (durch eine kfG gegebene) kontextfreie Sprache regulär ?

Zugehörige Reduktionsabbildungen

- Für die ersten drei Reduktionen verwende die Reduktionsabbildung

$$K \mapsto (M_1[K], M_2[K]) \text{ bzw. } K \mapsto (G_1[K], G_2[K]) \ .$$

- Für die letzten zwei Reduktionen verwende die Reduktionsabbildung $K \mapsto G'[K]$, wobei $G'[K]$ eine (aus K berechenbare !) kfG für die (kontextfreie !) Sprache $\overline{L_1[K]} \cup \overline{L_2[K]}$ ist.

Da die kfGs $G_1[K], G_2[K]$ eindeutig sind, bleiben die ersten drei Probleme der obigen Liste unentscheidbar, selbst wenn die beteiligten kontextfreien Sprachen durch eindeutige kfGs gegeben sind.

Weitere unentscheidbare Probleme

Inklusionsproblem: Gilt für zwei (durch DPDAs oder kfGs gegebene) deterministisch kontextfreie Sprachen L_1, L_2 die Beziehung $L_1 \subseteq L_2$?

Nachweis der Unentscheidbarkeit: Das Schnittproblem für deterministisch kontextfreie Sprachen

$$N(M_1) \cap N(M_2) = \emptyset \ ?$$

ist wegen

$$N(M_1) \cap N(M_2) = \emptyset \Leftrightarrow N(M_1) \subseteq \overline{N(M_2)}$$

auf das Inklusionsproblem reduzierbar. Als Reduktionsabbildung verwende

$$(M_1, M_2) \mapsto (M_1, M_2') \ ,$$

wobei M_2' ein (aus M_2 berechenbarer) DPDA für $\overline{N(M_2)}$ ist.

Weitere unentscheidbare Probleme (fortgesetzt)

Eindeutigkeitsproblem: Ist eine gegebene kfG eindeutig ?

Nachweis der Unentscheidbarkeit: Das Schnittproblem für eindeutige kontextfreie Sprachen

$$L(G_1) \cap L(G_2) = \emptyset \text{ } (G_1, G_2 \text{ eindeutige kfGs}) \text{ ?}$$

ist auf das Eindeutigkeitsproblem reduzierbar. Als Reduktionsabbildung verwende

$$(G_1, G_2) \mapsto G_3 \ ,$$

wobei G_3 die (aus G_1, G_2 mit der Standardtechnik berechenbare) kfG für die Sprache $L(G_1) \cup L(G_2)$ bezeichnet. Da G_1 und G_2 eindeutige kfGs sind, gibt es ein Wort mit zwei verschiedenen Syntaxbäumen über G_3 **gdw** $L(G_1) \cap L(G_2) \neq \emptyset$.

Weitere unentscheidbare Probleme (fortgesetzt)

Leerheit des Komplementes: Stimmt eine (durch eine kfG gegebene) kontextfreie Sprache mit Σ^* überein (d.h., sind alle Terminalstrings ableitbar) ?

Nachweis der Unentscheidbarkeit: Dieselbe Abbildung

$$(M_1, M_2) \mapsto G_3' \quad , \quad L(G_3') = \overline{N(M_1)} \cup \overline{N(M_2)} \ ,$$

wie in der vorangegangenen Reduktion reduziert das Schnittproblem für deterministisch kontextfreie Sprachen

$$N(M_1) \cap N(M_2) = \emptyset \ ?$$

auf das Problem der Leerheit des Komplementes.

Weitere unentscheidbare Probleme (fortgesetzt)

Äquivalenzproblem: Sind zwei gegebene kfGs äquivalent (d.h., sind die von ihnen erzeugten Sprachen identisch) ?

Nachweis der Unentscheidbarkeit: Das Problem der Leerheit des Komplementes

$$L(G) = \Sigma^* \ ?$$

ist auf das Äquivalenzproblem reduzierbar. Wenn G_* eine (leicht zu berechnende) kfG für Σ^* bezeichnet, dann kann als Reduktionsabbildung

$$G \mapsto (G, G_*)$$

verwendet werden.