VL-07: Das Postsche Correspondenzproblem

(Berechenbarkeit und Komplexität, WS 2018)

Gerhard Woeginger

WS 2018, RWTH

Organisatorisches

- Nächste Vorlesung:
  Freitag, November 23, 16:30–18:00 Uhr, Audimax
- Webseite:
  http://algo.rwth-aachen.de/Lehre/WS1819/BuK.php

Wiederholung

Wdh.: Entscheidbarkeit und Rekursive Aufzählbarkeit

Eine Sprache L ist rekursiv (oder entscheidbar), falls eine TM M existiert,

- die auf jeder Eingabe hält, und
- die genau die Wörter aus L akzeptiert.

Eine Sprache L ist semi-entscheidbar, falls eine TM M existiert,

- die jedes Wort aus L akzeptiert, und
- die kein Wort akzeptiert, das nicht in L enthalten ist.

Eine Sprache L ist rekursiv aufzählbar, falls ein Aufzähler A existiert,

- der genau die Worte in L druckt.

Es gilt: L semi-entscheidbar $\iff$ L rekursiv aufzählbar

Wdh.: Abschlusseigenschaften von Sprachen

Satz

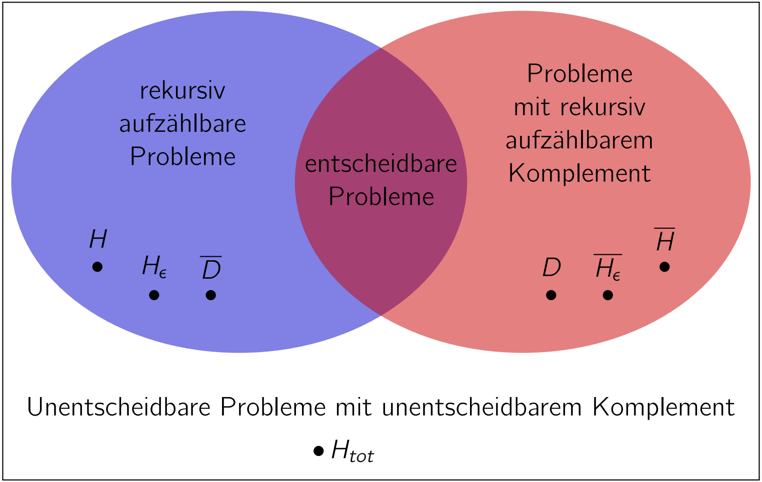
Die Menge der rekursiven Sprachen ist abgeschlossen unter Komplementbildung, Vereinigungen und Schnitten.

Satz

Die Menge der rekursiv aufzählbaren Sprachen

- ist abgeschlossen unter Vereinigungen und Schnitten
- ist nicht abgeschlossen unter Komplementbildung

Wdh.: Berechenbarkeitslandschaft



Wdh.: Reduktionen

Definition

Es seien L_1 und L_2 zwei Sprachen über einem Alphabet Σ.
Dann heisst L_1 auf L_2 reduzierbar (mit der Notation $L_1 \leq L_2$),
wenn eine berechenbare Funktion $f : \Sigma^* \to \Sigma^*$ existiert,
so dass für alle $x \in \Sigma^*$ gilt: $x \in L_1 \;\Leftrightarrow\; f(x) \in L_2$.

Satz

Es seien L_1 und L_2 zwei Sprachen mit $L_1 \leq L_2$.

- L_2 entscheidbar $\Rightarrow$ L_1 entscheidbar
- L_2 rekursiv aufzählbar $\Rightarrow$ L_1 rekursiv aufzählbar
- L_1 nicht entscheidbar $\Rightarrow$ L_2 nicht entscheidbar
- L_1 nicht rekursiv aufzählbar $\Rightarrow$ L_2 nicht rekursiv aufzählbar

Vorlesung VL-07
Das Postsche Correspondenzproblem

- Definition des PCP
- Definition des MPCP
- Unentscheidbarkeit von MPCP und PCP
- Leichte und schwierige Varianten

Das Postsche Correspondenzproblem (1)

Das Postsche Correspondenzproblem (PCP) ist ein Puzzle aus Dominos.

- Jeder Dominostein ist mit zwei Wörtern über einem Alphabet Σ beschriftet, und zwar mit einem Wort in der oberen Hälfte und einem Wort in der unteren Hälfte.
- Gegeben ist eine Menge K von Dominosteinen.
  Jeder Dominostein darf beliebig oft verwendet werden.
- Die Aufgabe besteht darin, eine correspondierende Folge von Dominos aus K zu finden, mit der sich oben und unten jeweils das selbe Wort ergibt.
- Die Folge muss mindestens einen Dominostein enthalten.

Das Postsche Correspondenzproblem (2)

Beispiel A

Für die Dominomenge

$$K \;=\; \left\{ \left[\frac{b}{ca}\right], \left[\frac{a}{ab}\right], \left[\frac{ca}{a}\right], \left[\frac{dbd}{cef}\right], \left[\frac{abc}{c}\right], \left[\frac{caeef}{abce}\right] \right\}$$

gibt es die correspondierende Folge $\langle 2, 1, 3, 2, 5 \rangle$ mit

$$\left[\frac{a}{ab}\right] \left[\frac{b}{ca}\right] \left[\frac{ca}{a}\right] \left[\frac{a}{ab}\right] \left[\frac{abc}{c}\right]$$

Das Postsche Correspondenzproblem (3)

Beispiel B

Nicht für jede Menge K existiert eine correspondierende Folge, wie zum Beispiel für die Dominomenge

$$K = \left\{ \left[\frac{abc}{ca}\right], \left[\frac{b}{aa}\right], \left[\frac{abcb}{abc}\right], \left[\frac{abc}{bc}\right] \right\}$$

Warum hat dieses PCP keine Lösung?

Als Übung können Sie versuchen, mit Computer Programmen die kürzeste Lösung für die folgenden drei PCPs zu finden:

Beispiel C

$$K_1 = \left\{ \left[\frac{aaba}{a}\right], \left[\frac{baab}{aa}\right], \left[\frac{a}{aab}\right] \right\}$$

$$K_2 = \left\{ \left[\frac{aaa}{aab}\right], \left[\frac{baa}{a}\right], \left[\frac{ab}{abb}\right], \left[\frac{b}{aa}\right] \right\}$$

$$K_3 = \left\{ \left[\frac{aab}{a}\right], \left[\frac{a}{ba}\right], \left[\frac{b}{aab}\right] \right\}$$

Das Postsche Correspondenzproblem (5)

Formale Definition (Postsches Correspondenzproblem, PCP)

Eine Instanz des PCP besteht aus einer endlichen Menge

$$K = \left\{ \left[\frac{x_1}{y_1}\right], \ldots, \left[\frac{x_k}{y_k}\right] \right\}$$

wobei $x_1, \ldots, x_k$ und $y_1, \ldots, y_k$ nichtleere Wörter über einem endlichen Alphabet Σ sind.

Das Problem besteht darin, zu entscheiden, ob es eine (nicht-leere) correspondierende Folge $\langle i_1, \ldots, i_n \rangle$ von Indizes in $\{1, \ldots, k\}$ gibt, sodass

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

Die Elemente von K nennen wir Dominosteine oder Dominos.

Emil Leon Post (1897–1954)

Wikipedia: Emil Post was a Polish-American mathematician and logician. He is best known for his work in the field that eventually became known as computability theory.

In 1936, Post developed, independently of Alan Turing, a mathematical model of computation which is sometimes called Post's machine or a Post-Turing machine.
Post's rewrite technique is now ubiquitous in programming language specification and design, and so with Church's lambda-calculus is a salient influence of classical modern logic on practical computing.

Das modifizierte PCP

Definition (Modifiziertes PCP; kurz: MPCP)

Eine Instanz des MPCP besteht aus einer endlichen Menge

$$K = \left\{ \left[\frac{x_1}{y_1}\right], \ldots, \left[\frac{x_k}{y_k}\right] \right\}$$

wobei $x_1, \ldots, x_k$ und $y_1, \ldots, y_k$ nicht-leere Wörter über einem endlichen Alphabet Σ sind.

Das Problem besteht darin zu entscheiden, ob es eine correspondierende Folge $\langle i_1, \ldots, i_n \rangle$ von Indizes $\boxed{\textbf{mit } \mathbf{i_1 = 1}}$ gibt, sodass gilt:

$$x_{i_1} x_{i_2} \ldots x_{i_n} \quad = \quad y_{i_1} y_{i_2} \ldots y_{i_n}$$

(Die Modifikation besteht also nur darin, dass der Stein $\left[\frac{x_1}{y_1}\right]$ das vorgegebene Startdomino ist, mit dem die correspondierende Folge beginnen muss.)

Arbeitsplan

Unser Arbeitsplan

Wir werden die folgenden beiden Aussagen beweisen.

Satz A

$MPCP \leq PCP$

Satz B

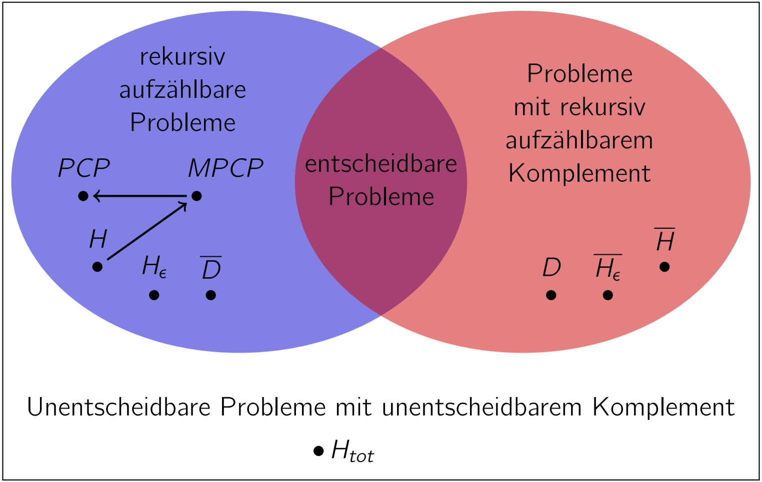
$H \leq MPCP$

Die Transitivität der Reduzierbarkeit (Übung) impliziert $H \leq PCP$.

Satz

Das PCP ist unentscheidbar.

Berechenbarkeitslandschaft



Beweis von Satz A: MPCP $\leq$ PCP

Beweis von $MPCP \le PCP$ (1)

Wir modellieren eine MPCP Instanz als PCP Instanz:

- Wir betrachten MPCP Instanz $K = \left\{ \left[\frac{x_1}{y_1}\right], \ldots, \left[\frac{x_k}{y_k}\right] \right\}$
- Es seien $\#$ und $\$$ zwei Symbole, die nicht im MPCP vorkommen
- Wir konstruieren x'_i aus x_i,
  indem wir hinter jedem Buchstaben ein $\#$ einfügen
- Wir konstruieren y'_i aus y_i,
  indem wir vor jedem Buchstaben ein $\#$ einfügen
- Ferner setzen wir $x'_0 = \#x'_1$; $y'_0 = y'_1$; $x'_{k+1} = \$$; und $y'_{k+1} = \#\$$.
- Damit berechnen wir die folgende PCP Instanz

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

Beweis von $MPCP \leq PCP$ (2)

Beispiel:

MPCP $K = \left\{ \left[\frac{ab}{a}\right], \left[\frac{c}{abc}\right], \left[\frac{a}{b}\right] \right\}$ wird modelliert als

PCP $f(K) = \left\{ \left[\frac{\#a\#b\#}{\#a}\right], \left[\frac{a\#b\#}{\#a}\right], \left[\frac{c\#}{\#a\#b\#c}\right], \left[\frac{a\#}{\#b}\right], \left[\frac{\$}{\#\$}\right] \right\}$

Lösung des MPCP:

$$\left[\frac{ab}{a}\right] \left[\frac{a}{b}\right] \left[\frac{ab}{a}\right] \left[\frac{c}{abc}\right]$$

Entsprechende Lösung des PCP:

$$\left[\frac{\#a\#b\#}{\#a}\right] \left[\frac{a\#}{\#b}\right] \left[\frac{a\#b\#}{\#a}\right] \left[\frac{c\#}{\#a\#b\#c}\right] \left[\frac{\$}{\#\$}\right]$$

Beweis von $MPCP \leq PCP$, Korrektheit (1)

Wir zeigen: (1) Wenn $K \in MPCP$, dann $f(K) \in PCP$

- Es sei $(i_1, i_2, \ldots, i_n)$ Lösung für MPCP K. Dann gilt $i_1 = 1$ und

$$x_{i_1} x_{i_2} \ldots x_{i_n} \;=\; y_{i_1} y_{i_2} \ldots y_{i_n} \;=\; a_1 a_2 \ldots a_s$$

  für gewisse Buchstaben $a_1, \ldots, a_s$ aus Σ.
- Dann ist $(0, i_2, \ldots, i_n, k+1)$ eine Lösung für PCP $f(K)$, denn

$$x'_0 x'_{i_2} \ldots x'_{i_n}\$ \;=\; \#a_1\#a_2\# \ldots \#a_s\#\$ \;=\; y'_0 y'_{i_2} \ldots y'_{i_n}\#\$$$

- Aus einer Lösung für MPCP K lässt sich also eine Lösung für PCP $f(K)$ konstruieren. Damit ist die Implikation (1) gezeigt.

Beweis von $MPCP \leq PCP$, Korrektheit (2)

Wir zeigen: (2) Wenn $f(K) \in PCP$, dann $K \in MPCP$

- Es sei $(i_1, i_2, \ldots, i_n)$ eine Lösung minimaler Länge für $f(K)$.
- Fakt A: $i_1 = 0$, da nur x'_0 und y'_0 mit dem selben Zeichen beginnen
- Fakt B: $i_n = k+1$, da nur x'_{k+1} und y'_{k+1} mit selbem Zeichen enden
- Fakt C: $i_j \neq k+1$ für $1 \leq j < n$. Andernfalls kürzere Lösung.
- Fakt D: $i_j \neq 0$ für $2 \leq j \leq n$. Andernfalls folgen im oberen Wort zwei $\#$-Zeichen direkt aufeinander, was im unteren Wort unmöglich ist.
- Durch das Löschen aller $\#$ und $\$$ Symbole wird das PCP Lösungswort also zum MPCP Lösungswort. $\square$

Beweis von Satz B: H $\le$ MPCP

Illustrierendes Beispiel (1)

Wir betrachten die TM $M = (Q, \Sigma, \Gamma, B, q_0, \overline{q}, \delta)$

- mit $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, B\}$, und $Q = \{q_0, q_1, q_2, \overline{q}\}$,
- und mit der folgenden Überführungsfunktion δ:

δ	0	1	B
q_0	$(q_0, 0, R)$	$(q_1, 1, R)$	$(\overline{q}, 1, N)$
q_1	$(q_2, 0, R)$	$(q_1, 1, R)$	$(\overline{q}, 1, N)$
q_2	$(q_2, 0, R)$	$(q_2, 1, R)$	(q_2, B, R)

Diese TM M erkennt die Sprache 0^*1^*:

- Bei Eingabeworten in 0^*1^* erreicht die Berechnung den Zustand $\overline{q}$, und die Maschine akzeptiert.
- Bei Eingabeworten nicht in 0^*1^* bleibt die Berechnung im Zustand q_2 stecken, und der Kopf läuft weiter und weiter nach rechts.

Illustrierendes Beispiel (2)

Die Berechnung der TM M auf einem gegebenen Eingabewort wird durch eine Konfigurationsfolge beschrieben:

Konfigurationsfolge von M auf Eingabe $w = 0011$

$$q_0 0011 \vdash 0 q_0 011 \vdash 00 q_0 11 \vdash 001 q_1 1 \vdash 0011 q_1 B \vdash 0011 \overline{q} 1$$

Wir werden solche Konfigurationsfolgen nun durch geeignet gewählte Dominos in einem MPCP beschreiben, kodieren, und simulieren.

Dominosteine / Teil 1

Beim Startdomino besteht das untere Wort aus der Anfangskonfiguration mit drei zusätzlichen Trennzeichen:

$$\left[\frac{\#}{\#\# q_0 0011 \#}\right]$$

Illustrierendes Beispiel (3)

Dominosteine / Teil 2

Weiters gibt es für jedes Zeichen aus $\Gamma \cup \{\#\}$ einen entsprechenden Stein:

$$\left[\frac{0}{0}\right], \left[\frac{1}{1}\right], \left[\frac{B}{B}\right], \left[\frac{\#}{\#}\right]$$

Dominosteine / Teil 3

Auch für jeden Eintrag in der Tabelle der Überführungsfunktion δ gibt es einen entsprechenden Stein, der den jeweiligen Übergang inklusive der Kopfbewegung beschreibt:

$$\left[\frac{q_0 0}{0 q_0}\right], \left[\frac{q_0 1}{1 q_1}\right], \left[\frac{q_0 B}{\overline{q} 1}\right], \left[\frac{q_1 0}{0 q_2}\right], \left[\frac{q_1 1}{1 q_1}\right], \left[\frac{q_1 B}{\overline{q} 1}\right], \left[\frac{q_2 0}{0 q_2}\right], \left[\frac{q_2 1}{1 q_2}\right], \left[\frac{q_2 B}{B q_2}\right]$$

(Achtung: Die Konstruktion wird später noch erweitert und fortgesetzt.)

Illustrierendes Beispiel (4)

Beobachtung:

Angenommen, wir ergänzen den Startdomino $\left[\frac{\#}{\#\#q_0 0011\#}\right]$ mit einer Folge von Dominos aus der bisherigen Liste erlaubter Dominos derart, dass der obere String einen Präfix des unteren Strings bildet.

- In der ersten Ergänzungsphase konstruieren wir dadurch im unteren String die Nachfolge-Konfiguration von M für $q_0 0011$.
- In den späteren Ergänzungsphasen konstruieren wir weitere Nachfolge-Konfigurationen, wobei der obere String dem unteren String immer um genau eine Konfiguration nachhinkt.

Illustrierendes Beispiel (5)

Rekonstruktion der Konfigurationsfolge

Die ersten Dominos in der Lösung des Puzzles sind

$$
\begin{array}{l}
\left[\dfrac{\#}{\#\#q_0 0011\#}\right] \left[\dfrac{\#}{\#}\right] \left[\dfrac{q_0 0}{0 q_0}\right] \left[\dfrac{0}{0}\right] \left[\dfrac{1}{1}\right] \left[\dfrac{1}{1}\right] \left[\dfrac{\#}{\#}\right] \\
\qquad\qquad\qquad\quad \left[\dfrac{\#}{\#}\right] \left[\dfrac{0}{0}\right] \left[\dfrac{q_0 0}{0 q_0}\right] \left[\dfrac{1}{1}\right] \left[\dfrac{1}{1}\right] \left[\dfrac{\#}{\#}\right] \\
\qquad\qquad\qquad\quad \left[\dfrac{\#}{\#}\right] \left[\dfrac{0}{0}\right] \left[\dfrac{0}{0}\right] \left[\dfrac{q_0 1}{1 q_1}\right] \left[\dfrac{1}{1}\right] \left[\dfrac{\#}{\#}\right] \\
\qquad\qquad\qquad\quad \left[\dfrac{\#}{\#}\right] \left[\dfrac{0}{0}\right] \left[\dfrac{0}{0}\right] \left[\dfrac{1}{1}\right] \left[\dfrac{q_1 1}{1 q_1}\right] \left[\dfrac{\#}{\#}\right] \\
\qquad\qquad\qquad\quad \left[\dfrac{\#}{\#}\right] \left[\dfrac{0}{0}\right] \left[\dfrac{0}{0}\right] \left[\dfrac{1}{1}\right] \left[\dfrac{1}{1}\right] \left[\dfrac{q_1 \#}{\overline{q}\, 1\#}\right] \quad \cdots \quad \cdots
\end{array}
$$

Illustrierendes Beispiel (6)

Alarm! Alarm! Alarm!

Der letzte Schritt war illegal,
da er einen nicht definierten Dominostein verwendet.

Deshalb ergänzen wir nun die Liste erlaubter Dominos:

Dominosteine / Teil 4

Die folgenden Dominos realisieren Überführungen, die ein zusätzliches Blank-Symbol benötigen, da der Kopf am Ende des Wortes steht.

$$\left[\frac{q_0\#}{\overline{q}\,1\#}\right], \quad \left[\frac{q_1\#}{\overline{q}\,1\#}\right]$$

Illustrierendes Beispiel (7)

Wie beenden wir die Geschichte nun?
Wie ermöglichen wir es dem oberen String, seinen ewigen Rückstand am Ende der Rechnung doch noch aufzuholen?

Dominosteine / Teil 5

Wir führen einige Dominos ein, die nur dann zum Einsatz kommen können, wenn der Endzustand $\overline{q}$ bereits erreicht ist:

$$\left[\frac{\overline{q}0}{\overline{q}}\right], \left[\frac{\overline{q}1}{\overline{q}}\right], \left[\frac{\overline{q}B}{\overline{q}}\right], \left[\frac{0\overline{q}}{\overline{q}}\right], \left[\frac{1\overline{q}}{\overline{q}}\right], \left[\frac{B\overline{q}}{\overline{q}}\right]$$

Schlussendlich fügen wir noch den Abschlussdomino hinzu:

$$\left[\frac{\#\overline{q}\#\#}{\#}\right]$$

Illustrierendes Beispiel (8)

Rekonstruktion der Konfigurationsfolge / Fortsetzung

$$\cdots \quad \left[\frac{\#}{\#}\right] \left[\frac{0}{0}\right] \left[\frac{0}{0}\right] \left[\frac{1}{1}\right] \left[\frac{1}{1}\right] \left[\frac{q_1\#}{\overline{q}1\#}\right]$$

$$\left[\frac{\#}{\#}\right] \left[\frac{0}{0}\right] \left[\frac{0}{0}\right] \left[\frac{1}{1}\right] \left[\frac{1}{1}\right] \left[\frac{\overline{q}1}{\overline{q}}\right] \left[\frac{\#}{\#}\right]$$

$$\left[\frac{\#}{\#}\right] \left[\frac{0}{0}\right] \left[\frac{0}{0}\right] \left[\frac{1}{1}\right] \left[\frac{1\overline{q}}{\overline{q}}\right] \left[\frac{\#}{\#}\right]$$

$$\left[\frac{\#}{\#}\right] \left[\frac{0}{0}\right] \left[\frac{0}{0}\right] \left[\frac{1\overline{q}}{\overline{q}}\right] \left[\frac{\#}{\#}\right]$$

$$\left[\frac{\#}{\#}\right] \left[\frac{0}{0}\right] \left[\frac{0\overline{q}}{\overline{q}}\right] \left[\frac{\#}{\#}\right]$$

$$\left[\frac{\#}{\#}\right] \left[\frac{0\overline{q}}{\overline{q}}\right] \left[\frac{\#}{\#}\right] \qquad \left[\frac{\#\overline{q}\#\#}{\#}\right] \qquad \text{(Fertig)}$$

Zurück zum Beweis von Satz B

Nach dem illustrierenden Beispiel kehren wir zum Beweis von Satz B zurück und beweisen die Aussage $H \leq MPCP$.

- Wir beschreiben eine berechenbare Funktion f, die eine syntaktisch korrekte Instanz $\langle M \rangle w$ fürs Halteproblem H in eine syntaktisch korrekte Instanz $K := f\big(\langle M \rangle w\big)$ fürs MPCP übersetzt
- Dabei gilt: M hält auf w $\iff$ K hat correspondierende Folge
- Syntaktisch nicht korrekte Eingaben für H werden auf syntaktisch nicht korrekte Eingaben fürs MPCP abgebildet
- Für die MPCP Instanz verwenden wir das Alphabet $\Gamma \cup Q \cup \{\#\}$ mit $\# \notin \Gamma \cup Q$

Die Reduktion (1)

Dominosteine (Startdomino)

Der Startdomino ist von der Form

$$\left[\frac{\#}{\#\#q_0w\#}\right]$$

Dominosteine (Kopierdominos)

Weiters enthält K die folgenden Kopierdominos:

$$\left[\frac{a}{a}\right] \text{ für alle } a \in \Gamma \cup \{\#\}$$

Die Reduktion (2)

Dominosteine (Überführungsdominos)

$$\left[\frac{qa}{q'c}\right] \quad \text{falls} \;\; \delta(q,a) = (q', c, N), \text{ für } q \in Q \setminus \{\overline{q}\},\; a \in \Gamma$$

$$\left[\frac{qa}{cq'}\right] \quad \text{falls} \;\; \delta(q,a) = (q', c, R), \text{ für } q \in Q \setminus \{\overline{q}\},\; a \in \Gamma$$

$$\left[\frac{bqa}{q'bc}\right] \quad \text{falls} \;\; \delta(q,a) = (q', c, L), \text{ für } q \in Q \setminus \{\overline{q}\},\; a, b \in \Gamma$$

Die Reduktion (3)

Dominosteine (Überführungsdominos für implizite Blanks)

$$\left[\frac{\#qa}{\#q'Bc}\right] \quad \text{falls } \delta(q,a) = (q',c,L), \text{ für } q \in Q \setminus \{\overline{q}\},\ a \in \Gamma$$

$$\left[\frac{q\#}{q'c\#}\right] \quad \text{falls } \delta(q,B) = (q',c,N), \text{ für } q \in Q \setminus \{\overline{q}\}$$

$$\left[\frac{q\#}{cq'\#}\right] \quad \text{falls } \delta(q,B) = (q',c,R), \text{ für } q \in Q \setminus \{\overline{q}\}$$

$$\left[\frac{bq\#}{q'bc\#}\right] \quad \text{falls } \delta(q,B) = (q',c,L), \text{ für } q \in Q \setminus \{\overline{q}\},\ b \in \Gamma$$

$$\left[\frac{\#q\#}{\#q'Bc\#}\right] \quad \text{falls } \delta(q,B) = (q',c,L), \text{ für } q \in Q \setminus \{\overline{q}\}$$

Die Reduktion (4)

Dominosteine (Löschdominos)

Weiters enthält K die folgenden Löschdominos:

$$\left[\frac{a\overline{q}}{\overline{q}}\right] \quad \text{und} \quad \left[\frac{\overline{q}a}{\overline{q}}\right] \quad \text{für } a \in \Gamma$$

Dominosteine (Abschlussdomino)

Der Abschlussdomino ist von der Form

$$\left[\frac{\#\overline{q}\#\#}{\#}\right]$$

Die Reduktion und die Beschreibung der Funktion f sind damit abgeschlossen.

Korrektheitsargument

Das Korrektheitsargument besteht aus drei Teilen:

Teil 1: f ist berechenbar (ist bereits erledigt)

Teil 2: M hält auf w $\Rightarrow$ $K \in MPCP$

Teil 3: $K \in MPCP$ $\Rightarrow$ M hält auf w

Korrektheit, Teil 2

Beweis von Teil 2: M hält auf $w \;\Rightarrow\; K \in MPCP$

- Die Berechnung von M auf w entspricht einer endlichen Konfigurationsfolge $k_0 \vdash k_1 \vdash \cdots \vdash k_{t-1} \vdash k_t$ wobei k_0 die Startkonfiguration im Zustand q_0 und k_t die Endkonfiguration im Zustand $\overline{q}$ ist.
- Wir konstruieren eine correspondierende Folge, die mit dem Startdomino beginnt.
  Der obere String ist ein Präfix des unteren Strings:
  $\#\# k_0 \#\# k_1 \#\# \cdots \#\# k_{t-1} \#$
  Der untere String gibt die vollständige Konfigurationsfolge an:
  $\#\# k_0 \#\# k_1 \#\# \cdots \#\# k_{t-1} \#\# k_t \#$
- Durch Hinzufügen von einer Folge von Löschdominos kann das Nachhinken des oberen Strings fast ausgeglichen werden. Danach sind beide Strings identisch bis auf einen Suffix der Form $\#\overline{q}\#$, der im oberen String fehlt.
- Hinzufügen des Abschlussdominos macht beide Strings identisch.

Korrektheit, Teil 3

Beweis von Teil 3: $K \in MPCP \;\Rightarrow\; M$ hält auf w

Der Satz von Dominosteinen im MPCP hat folgende Eigenschaften:

- Beim Startdomino ist der obere String kürzer als der untere
- Bei den Kopier- und Überführungsdominos ist der obere String immer höchstens so lang wie der untere String
- Nur auf Abschluss- und Löschdominos ist der obere String länger als der untere String

Die correspondierende Folge für K liefert uns eine entsprechende Konfigurationsfolge von M auf w.

- Diese Konfigurationsfolge beginnt mit dem Startdomino
- Diese Konfigurationsfolge muss zumindest einen Lösch- oder Abschlussdomino enthalten (andernfalls wäre der untere String länger als der obere String)
- Deshalb erscheint der Zustand $\overline{q}$ in dieser Konfigurationsfolge, und die Rechnung von M auf w terminiert

Leichte und schwierige Varianten des PCPs

Varianten des PCPs (1)

Wie ist die Komplexität für eingeschränkte Varianten des Problems?

Falls nur kurze Wörter erlaubt sind:

- Wenn alle Wörter auf den Dominos Länge 1 haben, so ist das PCP entscheidbar.
- Wenn alle Wörter Länge 1 oder 2 haben, so ist das PCP unentscheidbar.

Varianten des PCPs (2)

Falls nur wenige Dominos erlaubt sind:

- Für 1 Domino ist das PCP trivial.
- Für 2 Dominos ist das PCP entscheidbar.
  [Ehrenfeucht & Rozenberg] (1981)
- Für 3 und 4 Dominos ist die Komplexität ungeklärt.
- Für 5 Dominos ist das PCP unentscheidbar. [Neary] (2015)
- Für 7 Dominos oder mehr ist das PCP unentscheidbar.
  [Matijasevich & Sénizergues] (1996)
- Für unbeschränkt viele Dominos ist das PCP unentscheidbar.
  [Post] (1947)