Script generated by TTT

Title: Nipkow: Theo (24.06.2019)

Date: Mon Jun 24 14:18:55 CEST 2019

Duration: 90:46 min

Pages: 82

Definition 5.40
Eine Menge A ($\subseteq \mathbb{N}$ oder Σ^*) heißt semi-entscheidbar (s-e) gdw

$$\chi'_A(x) := \begin{cases} 1 & \text{falls } x \in A \\ \bot & \text{falls } x \notin A \end{cases}$$

berechenbar ist.

undefiniert

Satz 5.41
Eine Menge A ist entscheidbar gdw sowohl A als auch $\overline{A}$ s-e sind.

Satz 5.41

Eine Menge A ist entscheidbar gdw sowohl A als auch $\overline{A}$ s-e sind.

Beweis:

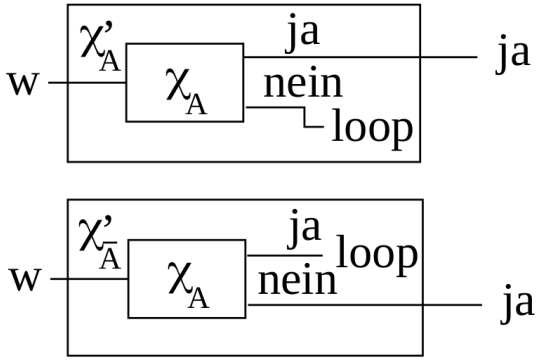
„$\Rightarrow$“: Wandle TM für χ_A in TM für χ'_A und $\chi'_{\overline{A}}$ um:

Satz 5.41

Eine Menge A ist entscheidbar gdw sowohl A als auch $\overline{A}$ s-e sind.

Beweis:

„$\Rightarrow$“: Wandle TM für χ_A in TM für χ'_A und $\chi'_{\overline{A}}$ um:



281

Beweis (Forts.):

„$\Leftarrow$“:
Wandle TM M_1 für χ'_A und TM M_2 für $\chi'_{\overline{A}}$ in TM für χ_A um:

input(x);
for $s := 0, 1, 2, \ldots$ **do**
  if $M_1[x]$ hält in s Schritten **then** output(1); **halt fi** ;
  if $M_2[x]$ hält in s Schritten **then** output(0); **halt fi**

Beweis (Forts.):

„$\Leftarrow$“:
Wandle TM M_1 für χ'_A und TM M_2 für $\chi'_{\overline{A}}$ in TM für χ_A um:

input(x);
for $s := 0, 1, 2, \ldots$ **do**
  if $M_1[x]$ hält in s Schritten **then** output(1); **halt fi** ;
  if $M_2[x]$ hält in s Schritten **then** output(0); **halt fi**

Formulierung mit Parallelismus:

input(x);
führe $M_1[x]$ und $M_2[x]$ parallel aus;
hält M_1, gib 1 aus, hält M_2, gib 0 aus. □

282

Beweis (Forts.):

„⇐“:
Wandle TM M_1 für χ'_A und TM M_2 für $\chi'_{\overline{A}}$ in TM für χ_A um:

input(x);
for $s := 0, 1, 2, \ldots$ **do**
  if $M_1[x]$ hält in s Schritten **then** output(1); **halt fi** ;
  if $M_2[x]$ hält in s Schritten **then** output(0); **halt fi**

Formulierung mit Parallelismus:

input(x);
führe $M_1[x]$ und $M_2[x]$ parallel aus;
hält M_1, gib 1 aus, hält M_2, gib 0 aus. □

Lemma 5.42

Ist $A \leq B$ und ist B s-e, so ist auch A s-e.

Beweis: Übung

Definition 5.43

Eine Menge A heißt rekursiv aufzählbar (*recursively enumerable*) gdw $A = \emptyset$ oder es eine berechenbare totale Funktion $f : \mathbb{N} \to A$ gibt, so dass

$$A = \{f(0), f(1), f(2), \ldots\}$$

Bemerkung:

- Es dürfen Elemente doppelt auftreten ($f(i) = f(j)$ für $i \neq j$)
- Die Reihenfolge ist beliebig.

Definition 5.43

Eine Menge A heißt rekursiv aufzählbar (*recursively enumerable*) gdw $A = \emptyset$ oder es eine berechenbare totale Funktion $f : \mathbb{N} \to A$ gibt, so dass

$$A = \{f(0), f(1), f(2), \ldots\}$$

Bemerkung:

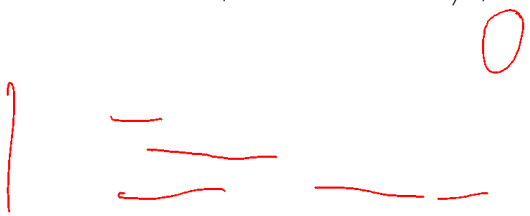
- Es dürfen Elemente doppelt auftreten ($f(i) = f(j)$ für $i \neq j$)
- Die Reihenfolge ist beliebig.

Lemma 5.44

Eine Menge A ist rekursiv aufzählbar gdw sie semi-entscheidbar ist.

Beweis:

Der Fall $A = \emptyset$ ist trivial. Sei $A \neq \emptyset$.



Lemma 5.44

Eine Menge A ist rekursiv aufzählbar gdw sie semi-entscheidbar ist.

Beweis:

Der Fall $A = \emptyset$ ist trivial. Sei $A \neq \emptyset$.

„$\Rightarrow$“: Sei A rekursiv aufzählbar mit f. Dann ist A semi-entscheidbar:

input(x);
for $i := 0, 1, 2, \ldots$ **do**
  if $f(i) = x$ **then** output(1); **halt** **fi**

„$\Leftarrow$“: O.B.d.A. nehmen wir $A \subseteq \mathbb{N}$ an.

Lemma 5.44

Eine Menge A ist rekursiv aufzählbar gdw sie semi-entscheidbar ist.

Beweis:

Der Fall $A = \emptyset$ ist trivial. Sei $A \neq \emptyset$.

„$\Rightarrow$“: Sei A rekursiv aufzählbar mit f. Dann ist A semi-entscheidbar:

input(x);
for $i := 0, 1, 2, \ldots$ **do**
  if $f(i) = x$ **then** output(1); **halt** **fi**

„$\Leftarrow$“: O.B.d.A. nehmen wir $A \subseteq \mathbb{N}$ an.
Sei A semi-entscheidbar durch (zB) GOTO-Programm P.

Lemma 5.44

Eine Menge A ist rekursiv aufzählbar gdw sie semi-entscheidbar ist.

Beweis:

Der Fall $A = \emptyset$ ist trivial. Sei $A \neq \emptyset$.

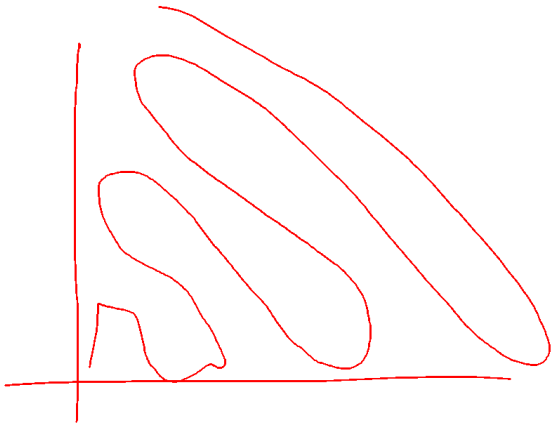
„$\Rightarrow$“: Sei A rekursiv aufzählbar mit f. Dann ist A semi-entscheidbar:

input(x);
for $i := 0, 1, 2, \ldots$ **do**
  if $f(i) = x$ **then** output(1); **halt** **fi**

„$\Leftarrow$“: O.B.d.A. nehmen wir $A \subseteq \mathbb{N}$ an.
Sei A semi-entscheidbar durch (zB) GOTO-Programm P.
Problem: $P[i]$ muss nicht halten und darf daher nur „zeitbeschränkt“ ausgeführt werden.

Beweis (Forts.):

Idee: Wir benutzen eine geeignete Bijektion $c\colon \mathbb{N} \times \mathbb{N} \leftrightarrow \mathbb{N}$.



Beweis (Forts.):

Idee: Wir benutzen eine geeignete Bijektion $c\colon \mathbb{N} \times \mathbb{N} \leftrightarrow \mathbb{N}$.
Seien $p_1\colon \mathbb{N} \to \mathbb{N}$ und $p_2\colon \mathbb{N} \to \mathbb{N}$ mit

$$p_1(c(n_1, n_2)) = n_1 \qquad \text{und} \qquad p_2(c(n_1, n_2)) = n_2$$

(Umkehrung von c).

Beweis (Forts.):

Idee: Wir benutzen eine geeignete Bijektion $c\colon \mathbb{N} \times \mathbb{N} \leftrightarrow \mathbb{N}$.
Seien $p_1\colon \mathbb{N} \to \mathbb{N}$ und $p_2\colon \mathbb{N} \to \mathbb{N}$ mit

$$p_1(c(n_1, n_2)) = n_1 \qquad \text{und} \qquad p_2(c(n_1, n_2)) = n_2$$

(Umkehrung von c).

Sei $d \in A$ beliebig.

Beweis (Forts.):

Idee: Wir benutzen eine geeignete Bijektion $c\colon \mathbb{N} \times \mathbb{N} \leftrightarrow \mathbb{N}$.
Seien $p_1\colon \mathbb{N} \to \mathbb{N}$ und $p_2\colon \mathbb{N} \to \mathbb{N}$ mit

$$p_1(c(n_1, n_2)) = n_1 \qquad \text{und} \qquad p_2(c(n_1, n_2)) = n_2$$

(Umkehrung von c).

Sei $d \in A$ beliebig.
Folgender Algorithmus berechnet eine Aufzählung von A:

input(n);
if $P[p_1(n)]$ hält nach $p_2(n)$ Schritten **then** output($p_1(n)$)
else output(d) **fi**

Korrektheit: Der Algorithmus hält immer und liefert immer ein Element aus A.

Satz 5.45

Die Menge $K = \{w \mid M_w[w]\downarrow\}$ ist semi-entscheidbar.

Satz 5.45
Die Menge $K = \{w \mid M_w[w]\downarrow\}$ ist semi-entscheidbar.

Beweis:
Die Funktion χ'_K ist wie folgt Turing-berechenbar:

Bei Eingabe w simuliere die Ausführung von $M_w[w]$;
gib 1 aus. □

Satz 5.45
Die Menge $K = \{w \mid M_w[w]\downarrow\}$ ist semi-entscheidbar.

Beweis:
Die Funktion χ'_K ist wie folgt Turing-berechenbar:

Bei Eingabe w simuliere die Ausführung von $M_w[w]$;
gib 1 aus. □

- Hier haben wir benutzt, dass man einen Interpreter/Simulator für Turingmaschinen als Turingmaschine programmieren kann.
- Ein solcher Interpreter wird oft eine Universelle Turingmaschine (U) genannt.

Korollar 5.46
$\overline{K}$ ist nicht semi-entscheidbar.

Semi-Entscheidbarkeit ist nicht abgeschlossen unter Komplement.

287

5.7 Die Sätze von Rice und Shapiro

Die von der TM M_w berechnete Funktion bezeichnen wir mit φ_w.
Wir betrachten implizit nur einstellige Funktionen.

5.7 Die Sätze von Rice und Shapiro

Die von der TM M_w berechnete Funktion bezeichnen wir mit φ_w.
Wir betrachten implizit nur einstellige Funktionen.

Satz 5.47 (Rice)
Sei F eine Menge berechenbarer Funktionen.

288

5.7 Die Sätze von Rice und Shapiro

Die von der TM M_w berechnete Funktion bezeichnen wir mit φ_w.
Wir betrachten implizit nur einstellige Funktionen.

Satz 5.47 (Rice)

Sei F eine Menge berechenbarer Funktionen.
Es gelte weder $F = \emptyset$ noch $F =$ alle ber. Funkt. („F nicht trivial“)
Dann ist unentscheidbar, ob die von einer gegebenen TM M_w berechnete Funktion Element F ist, dh ob $\varphi_w \in F$.

5.7 Die Sätze von Rice und Shapiro

Die von der TM M_w berechnete Funktion bezeichnen wir mit φ_w.
Wir betrachten implizit nur einstellige Funktionen.

Satz 5.47 (Rice)

Sei F eine Menge berechenbarer Funktionen.
Es gelte weder $F = \emptyset$ noch $F =$ alle ber. Funkt. („F nicht trivial“)
Dann ist unentscheidbar, ob die von einer gegebenen TM M_w berechnete Funktion Element F ist, dh ob $\varphi_w \in F$.

Alle nicht-triviale semantische Eigenschaften von Programmen sind unentscheidbar.

5.7 Die Sätze von Rice und Shapiro

Die von der TM M_w berechnete Funktion bezeichnen wir mit φ_w.
Wir betrachten implizit nur einstellige Funktionen.

Satz 5.47 (Rice)

Sei F eine Menge berechenbarer Funktionen.
Es gelte weder $F = \emptyset$ noch $F =$ alle ber. Funkt. („F nicht trivial“)
Dann ist unentscheidbar, ob die von einer gegebenen TM M_w berechnete Funktion Element F ist, dh ob $\varphi_w \in F$.

Alle nicht-triviale semantische Eigenschaften von Programmen sind unentscheidbar.

Beispiel 5.48

Es ist unentscheidbar, ob ein Programm

- für mindestens eine Eingabe hält.
  $(F = \{\varphi_w \mid \exists x.\ M_w[x]\downarrow\})$

5.7 Die Sätze von Rice und Shapiro

Die von der TM M_w berechnete Funktion bezeichnen wir mit φ_w.
Wir betrachten implizit nur einstellige Funktionen.

Satz 5.47 (Rice)

Sei F eine Menge berechenbarer Funktionen.
Es gelte weder $F = \emptyset$ noch $F =$ alle ber. Funkt. („F nicht trivial“)
Dann ist unentscheidbar, ob die von einer gegebenen TM M_w berechnete Funktion Element F ist, dh ob $\varphi_w \in F$.

Alle nicht-triviale semantische Eigenschaften von Programmen sind unentscheidbar.

Beispiel 5.48

Es ist unentscheidbar, ob ein Programm

- für mindestens eine Eingabe hält.
  $(F = \{\varphi_w \mid \exists x.\ M_w[x]\downarrow\})$
- für alle Eingaben hält. $(F = \{\varphi_w \mid \forall x.\ M_w[x]\downarrow\})$

Satz 5.49 (Rice-Shapiro)

Sei F eine Menge berechenbarer Funktionen.
Ist $C_F := \{w \mid \varphi_w \in F\}$ semi-entscheidbar,
so gilt für alle berechenbaren f:
$f \in F \Leftrightarrow$ es gibt eine endliche Teilfunktion $g \subseteq f$ mit $g \in F$.

Satz 5.49 (Rice-Shapiro)

Sei F eine Menge berechenbarer Funktionen.
Ist $C_F := \{w \mid \varphi_w \in F\}$ semi-entscheidbar,
so gilt für alle berechenbaren f:
$f \in F \Leftrightarrow$ es gibt eine endliche Teilfunktion $g \subseteq f$ mit $g \in F$.

Beweis:
„$\Rightarrow$“ mit Widerspruch.

Satz 5.49 (Rice-Shapiro)

Sei F eine Menge berechenbarer Funktionen.
Ist $C_F := \{w \mid \varphi_w \in F\}$ semi-entscheidbar,
so gilt für alle berechenbaren f:
$f \in F \Leftrightarrow$ es gibt eine endliche Teilfunktion $g \subseteq f$ mit $g \in F$.

Beweis:
„$\Rightarrow$“ mit Widerspruch.
Sei $f \in F$, so dass für alle endlichen $g \subseteq f$ gilt $g \notin F$.

Satz 5.49 (Rice-Shapiro)

Sei F eine Menge berechenbarer Funktionen.
Ist $C_F := \{w \mid \varphi_w \in F\}$ semi-entscheidbar,
so gilt für alle berechenbaren f:
$f \in F \Leftrightarrow$ es gibt eine endliche Teilfunktion $g \subseteq f$ mit $g \in F$.

Beweis:
„$\Rightarrow$“ mit Widerspruch.
Sei $f \in F$, so dass für alle endlichen $g \subseteq f$ gilt $g \notin F$.
Wir zeigen $\overline{K} \leq C_F$ womit C_F nicht semi-entscheidbar ist.

Satz 5.49 (Rice-Shapiro)

Sei F eine Menge berechenbarer Funktionen.
Ist $C_F := \{w \mid \varphi_w \in F\}$ semi-entscheidbar,
so gilt für alle berechenbaren f:
$f \in F \Leftrightarrow$ es gibt eine endliche Teilfunktion $g \subseteq f$ mit $g \in F$.

Beweis:

„$\Rightarrow$“ mit Widerspruch.
Sei $f \in F$, so dass für alle endlichen $g \subseteq f$ gilt $g \notin F$.
Wir zeigen $\overline{K} \leq C_F$ womit C_F nicht semi-entscheidbar ist.
Widerspruch

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Wir zeigen

$$w \in \overline{K} \Leftrightarrow h(w) \in C_F$$

Beweis (Forts.):

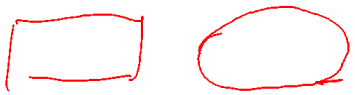
Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Wir zeigen

$$w \in \overline{K} \;\Leftrightarrow\; h(w) \in C_F$$

- $w \in \overline{K}$



Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Wir zeigen

$$w \in \overline{K} \;\Leftrightarrow\; h(w) \in C_F$$

- $w \in \overline{K} \implies \neg M_w[w]{\downarrow} \implies \varphi_{h(w)} = f \in F \implies h(w) \in C_F$

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Wir zeigen

$$w \in \overline{K} \;\Leftrightarrow\; h(w) \in C_F$$

- $w \in \overline{K} \implies \neg M_w[w]{\downarrow} \implies \varphi_{h(w)} = f \in F \implies h(w) \in C_F$
- Falls $w \notin \overline{K}$ dann hält $M_w[w]$ nach einer Zahl t von Schritten.

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Wir zeigen

$$w \in \overline{K} \;\Leftrightarrow\; h(w) \in C_F$$

- $w \in \overline{K} \implies \neg M_w[w]{\downarrow} \implies \varphi_{h(w)} = f \in F \implies h(w) \in C_F$
- Falls $w \notin \overline{K}$ dann hält $M_w[w]$ nach eine Zahl t von Schritten.
  Damit gilt: $\varphi_{h(w)}$ ist f eingeschränkt auf $\{0, \ldots, t-1\}$.

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Wir zeigen

$$w \in \overline{K} \;\Leftrightarrow\; h(w) \in C_F$$

- $w \in \overline{K} \implies \neg M_w[w]{\downarrow} \implies \varphi_{h(w)} = f \in F \implies h(w) \in C_F$
- Falls $w \notin \overline{K}$ dann hält $M_w[w]$ nach eine Zahl t von Schritten.
  Damit gilt: $\varphi_{h(w)}$ ist f eingeschränkt auf $\{0, \ldots, t-1\}$.
  Nach Annahme folgt $\varphi_{h(w)} \notin F$, dh $h(w) \notin C_F$.

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Wir zeigen

$$w \in \overline{K} \;\Leftrightarrow\; h(w) \in C_F$$

- $w \in \overline{K} \implies \neg M_w[w]{\downarrow} \implies \varphi_{h(w)} = f \in F \implies h(w) \in C_F$
- Falls $w \notin \overline{K}$ dann hält $M_w[w]$ nach eine Zahl t von Schritten.

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife, sonst berechne $f(t)$.

Beweis (Forts.):

„$\Leftarrow$“ mit Widerspruch.

Beweis (Forts.):

Reduktion $\overline{K} \leq C_F$ mit $h : \{0,1\}^* \to \{0,1\}^*$:
$h(w)$ ist die Kodierung folgender TM:

Bei Eingabe t simuliere t Schritte von $M_w[w]$.
Hält diese Berechnung in $\leq t$ Schritten, gehe in eine endlos Schleife,
sonst berechne $f(t)$.

Wir zeigen

$$w \in \overline{K} \;\Leftrightarrow\; h(w) \in C_F$$

Rice-Shapiro (in Kurzform): $C_F := \{w \mid \varphi_w \in F\}$ s-e $\Longrightarrow$

$$f \in F \Leftrightarrow \text{es gibt endliche Funkt. } g \subseteq f \text{ mit } g \in F.$$

Ein Programm heißt terminierend gdw es für alle Eingaben hält.

Korollar 5.50

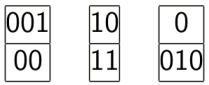
- *Die Menge der terminierenden Programme ist nicht semi-entscheidbar.*
- *Die Menge der nicht-terminierenden Programme ist nicht semi-entscheidbar.*

Beweis:

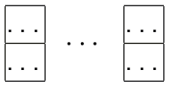
- $F :=$ Menge aller berechenbaren totalen Funktionen.

5.8 Das Postsche Korrespondenzproblem

Gegeben beliebig viele Kopien der 3 „Spielkarten“



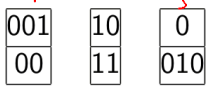
gibt es dann eine Folge dieser Karten



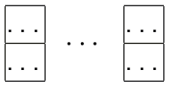
so dass oben und unten das gleiche Wort steht?

5.8 Das Postsche Korrespondenzproblem

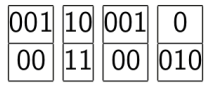
Gegeben beliebig viele Kopien der 3 „Spielkarten“



gibt es dann eine Folge dieser Karten



so dass oben und unten das gleiche Wort steht?



Kurz: 1,2,1,3.

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Problem: Gibt es eine Folge von Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$, $n > 0$, mit $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Folge von Indizes $\in \{1, \ldots, k\}$

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Problem: Gibt es eine Folge von Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$, $n > 0$, mit $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Dann nennen wir $i_1, \ldots, i_n$ eine Lösung der Instanz $(x_1, y_1), \ldots, (x_k, y_k)$ des PCP Problems.

Beispiel 5.52

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Problem: Gibt es eine Folge von Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$, $n > 0$, mit $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Dann nennen wir $i_1, \ldots, i_n$ eine Lösung der Instanz $(x_1, y_1), \ldots, (x_k, y_k)$ des PCP Problems.

Beispiel 5.52

- Hat $(1, 111), (10111, 10), (10, 0)$ eine Lösung? 2,1,1,3

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Problem: Gibt es eine Folge von Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$, $n > 0$, mit $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Dann nennen wir $i_1, \ldots, i_n$ eine Lösung der Instanz $(x_1, y_1), \ldots, (x_k, y_k)$ des PCP Problems.

Beispiel 5.52

- Hat $(1, 111), (10111, 10), (10, 0)$ eine Lösung? 2,1,1,3
- Hat $(b, ca), (a, ab), (ca, a), (abc, c)$ eine Lösung?

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Problem: Gibt es eine Folge von Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$, $n > 0$, mit $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Dann nennen wir $i_1, \ldots, i_n$ eine Lösung der Instanz $(x_1, y_1), \ldots, (x_k, y_k)$ des PCP Problems.

Beispiel 5.52

- Hat $(1, 111), (10111, 10), (10, 0)$ eine Lösung? 2,1,1,3
- Hat $(b, ca), (a, ab), (ca, a), (abc, c)$ eine Lösung? 2,1,3,2,4

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Problem: Gibt es eine Folge von Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$, $n > 0$, mit $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Dann nennen wir $i_1, \ldots, i_n$ eine Lösung der Instanz $(x_1, y_1), \ldots, (x_k, y_k)$ des PCP Problems.

Beispiel 5.52

- Hat $(1, 111), (10111, 10), (10, 0)$ eine Lösung? 2,1,1,3
- Hat $(b, ca), (a, ab), (ca, a), (abc, c)$ eine Lösung? 2,1,3,2,4
- Hat $(101, 01), (101, 010), (010, 10)$ eine Lösung? Nein!
- Hat $(10, 101), (011, 11), (101, 011)$ eine Lösung?

298

Definition 5.51 (Postsche Korrespondenzproblem, *Post's Correspondence Problem*, PCP)

Gegeben: Eine endliche Folge $(x_1, y_1), \ldots, (x_k, y_k)$, wobei $x_i, y_i \in \Sigma^+$.

Problem: Gibt es eine Folge von Indizes $i_1, \ldots, i_n \in \{1, \ldots, k\}$, $n > 0$, mit $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$?

Dann nennen wir $i_1, \ldots, i_n$ eine Lösung der Instanz $(x_1, y_1), \ldots, (x_k, y_k)$ des PCP Problems.

Beispiel 5.52

- Hat $(1, 111), (10111, 10), (10, 0)$ eine Lösung? 2,1,1,3
- Hat $(b, ca), (a, ab), (ca, a), (abc, c)$ eine Lösung? 2,1,3,2,4
- Hat $(101, 01), (101, 010), (010, 10)$ eine Lösung? Nein!
- Hat $(10, 101), (011, 11), (101, 011)$ eine Lösung? [HMU]
- Hat $(1000, 10), (1, 0011), (0, 111), (11, 0)$ eine Lösung?

298

Emil Post.
A Variant of a Recursively Unsolvable Problem.
Bulletin American Mathematical Society, 1946.

Emil Leon Post, 1897 (Polen) – 1954 (NY).

Lemma 5.53
Das PCP ist semi-entscheidbar.

Lemma 5.53
Das PCP ist semi-entscheidbar.

Beweis:
Zähle die möglichen Lösungen der Länge nach auf, und probiere jeweils, ob es eine wirkliche Lösung ist. □

Lemma 5.53
Das PCP ist semi-entscheidbar.

Beweis:
Zähle die möglichen Lösungen der Länge nach auf, und probiere jeweils, ob es eine wirkliche Lösung ist. □

Wir zeigen nun:

$$H \leq MPCP \leq PCP$$

Lemma 5.53
Das PCP ist semi-entscheidbar.

Beweis:
Zähle die möglichen Lösungen der Länge nach auf, und probiere jeweils, ob es eine wirkliche Lösung ist. □

Wir zeigen nun:

$$H \leq MPCP \leq PCP$$

wobei

Definition 5.54 (Modifiziertes PCP, MPCP)

Gegeben: wie beim PCP

Problem: Gibt es eine Lösung $i_1, \ldots, i_n$ mit $i_1 = 1$?

Satz 5.55
$MPCP \leq PCP$

Satz 5.55
$MPCP \leq PCP$

Beweis:
Für $w = a_1 \dots a_n$:

Satz 5.55
$MPCP \leq PCP$

Beweis:
Für $w = a_1 \dots a_n$:

$$\overline{w} := \#a_1\#a_2\#\dots\#a_n\#$$

Satz 5.55
$MPCP \leq PCP$

Beweis:
Für $w = a_1 \dots a_n$:

$$\begin{aligned} \overline{w} &:= \#a_1\#a_2\#\dots\#a_n\# \\ \overleftarrow{w} &:= a_1\#a_2\#\dots\#a_n\# \end{aligned}$$

Satz 5.55

$MPCP \leq PCP$

Beweis:

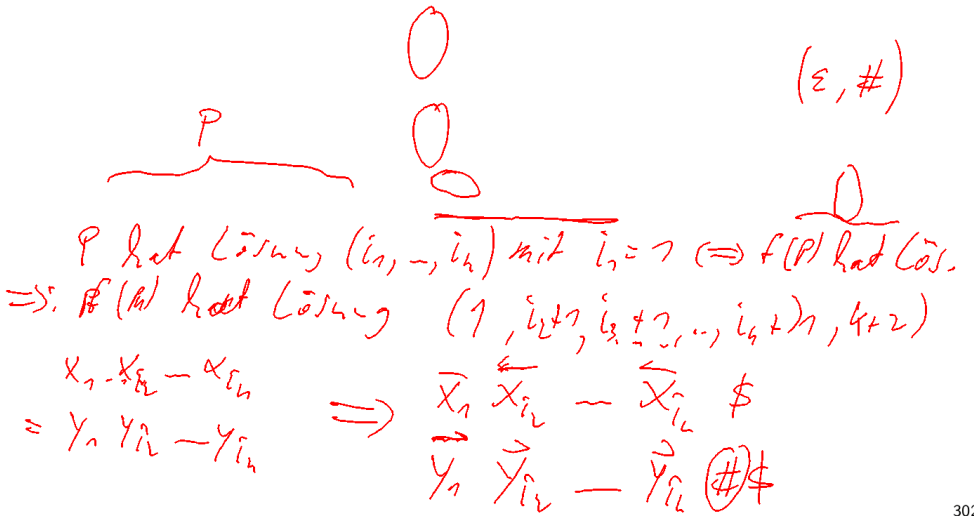
Für $w = a_1 \dots a_n$:

$$\begin{aligned} \overline{w} &:= \#a_1\#a_2\#\dots\#a_n\# \\ \overleftarrow{w} &:= a_1\#a_2\#\dots\#a_n\# \\ \overrightarrow{w} &:= \#a_1\#a_2\#\dots\#a_n \end{aligned}$$

$f((x_1, y_1), \dots, (x_k, y_k)) := ((\overline{x_1}, \overrightarrow{y_1}), (\overleftarrow{x_1}, \overrightarrow{y_1}), \dots, (\overleftarrow{x_k}, \overrightarrow{y_k}), (\$, \#\$))$

Satz 5.56

$H \leq MPCP$



Satz 5.56

$H \leq MPCP$

Satz 5.56

$H \leq MPCP$

Beweis:

Gesucht f:

Satz 5.56

$$H \leq MPCP$$

Beweis:

Satz 5.56

$$H \leq MPCP$$

Bsp $P = (1, 100), (001, 1)$ Lsg: 1001

$f(P) = (\#1\#, \#1\#0\#0), (0\#0\#1\#, \#1)$

$\#1\#$ $0\#0\#1\#$ $\$$

$\#1\#0\#0$ $\#1$ $\#\$$

Gesucht TM $M_w = (Q, \Sigma, \Gamma, \delta, q_1, \square, F)$

Eingabe w

$H \leq MPCP$

Beweis:

Satz 5.56

$H \leq MPCP$

Beweis:

- $(\#, \#q_0u\#)$
- (a, a) für alle $a \in \Gamma \cup \{\#\}$
- $(qa, q'a')$ falls $\delta(q, a) = (q', a', N)$
  $(qa, a'q')$ falls $\delta(q, a) = (q', a', R)$
  $(bqa, q'ba')$ falls $\delta(q, a) = (q', a', L)$, für alle $b \in \Gamma$
- $(\#, \square\#)$, $(\#, \#\square)$
- (aq, q), (qa, q) für alle $q \in F, a \in \Gamma$
- $(q\#\#, \#)$ für alle $q \in F$

Satz 5.56

$H \leq MPCP$

Beweis:

Aus $H \leq PCP$ folgt direkt

Korollar 5.57

Das PCP ist unentscheidbar.

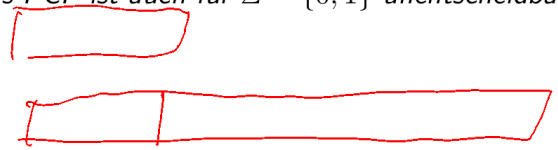
Aus $H \leq PCP$ folgt direkt

Korollar 5.57

Das PCP ist unentscheidbar.

Korollar 5.58

Das PCP ist auch für $\Sigma = \{0, 1\}$ unentscheidbar



Aus $H \leq PCP$ folgt direkt

Korollar 5.57
Das PCP ist unentscheidbar.

Korollar 5.58
Das PCP ist auch für $\Sigma = \{0, 1\}$ unentscheidbar

Beweis:
Wir nennen dies das 01-PCP und zeigen PCP $\leq$ 01-PCP.

Aus $H \leq PCP$ folgt direkt

Korollar 5.57
Das PCP ist unentscheidbar.

Korollar 5.58
Das PCP ist auch für $\Sigma = \{0, 1\}$ unentscheidbar

Beweis:
Wir nennen dies das 01-PCP und zeigen PCP $\leq$ 01-PCP.
Sei $\Sigma = \{a_1, \ldots, a_m\}$ das Alphabet des gegebenen PCPs.

Aus $H \leq PCP$ folgt direkt

Korollar 5.57
Das PCP ist unentscheidbar.

Korollar 5.58
Das PCP ist auch für $\Sigma = \{0, 1\}$ unentscheidbar

Beweis:
Wir nennen dies das 01-PCP und zeigen PCP $\leq$ 01-PCP.
Sei $\Sigma = \{a_1, \ldots, a_m\}$ das Alphabet des gegebenen PCPs.

Abbildung auf ein 01-PCP:

$$\begin{aligned} \widehat{a_j} &:= 01^j \\ \widehat{a_{j_1} \ldots a_{j_n}} &:= \widehat{a_{j_1}} \ldots \widehat{a_{j_n}} \end{aligned}$$