Große Übung 4: Reduktionen + Zählermaschinen

In den HA: $HP_\varepsilon \leq$ Totality

$HP_\varepsilon \leq \overline{\text{Totality}}$: Gesucht ist f: Instances(HP_ε) $\longrightarrow$ Instances($\overline{\text{Totality}}$)

$\Updownarrow$ $\qquad$ Instances(HP_ε) = TM $\qquad$ Instances($\overline{\text{Totality}}$) = Instances(Totality) = TM

$\overline{HP_\varepsilon} \leq$ Totality

nehme dazu das Idee: (Eingabe-TM M_x)

$$L(M_{f(x)}) = \Sigma^* \setminus \{\langle c\rangle \mid c \text{ ist akzeptierende Berechnung von } M_x \text{ auf } \varepsilon\}$$

Falls $\varepsilon \in L(M_x)$, dann gibt es so ein $\langle c\rangle$, also $L(M_{f(x)}) = \Sigma^* \setminus \{\langle c\rangle\} \neq \Sigma^*$

Anderenfalls $\varepsilon \notin L(M_x)$, dann gibt es $\langle c\rangle$ nicht, also $L(M_{f(x)}) = \Sigma^* \setminus \emptyset$

(Wäre Totality semi-entscheidbar, dann könne man eine TM für $\overline{HP_\varepsilon}$ konstruieren:

$$x \longrightarrow \boxed{M_f} \xrightarrow{f(x)} \boxed{M_{\text{Totality}}} \xrightarrow{acc} \circledcirc$$

M_f existiert, z.B. gibt es M', die den folgenden Algo implementiert:

Eingabe: $x\#y \in \{0,1\}^*\#\{0,1\}^*$ (Idee: Teste ob y ~~eine~~ keine akzeptierende Berechnung von M_x auf ε ist)

1. Falls $y \neq \langle c_0, c_1, \dots, c_n\rangle$, ~~loope unendlich / reject~~ accept.
2. Falls c_0 nicht $\langle \varepsilon, q_0^{\text{von } M_x}, \sqcup\rangle$, ~~reject~~ accept

3. Falls c_n nicht $\Gamma^* \times Q_F \times \Gamma^*$, ~~reject~~ accept

4. Für $i = 0 \ldots n-1$

4.1. Falls c_{i+1} keine Folgekonf. von M_x von c_i ist, ~~reject~~ accept

end

c_i: i-te Konfiguration aus der Folge, die durch y kodiert wird

5. ~~accept~~ loope unendlich

$$L(M') = \{ x\#y \mid y \text{ ist keine akz. Berechnung von } M_x \text{ auf } \varepsilon \}$$

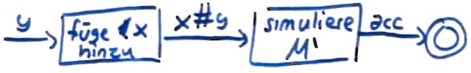
Nach Church-Turing gibt es M'. Es gibt also $\langle M' \rangle$.

M_f ~~mach~~ soll den folgenden Algo implementieren:

<u>Eingabe</u>: x

'gib' $f(x)$ aus mit

$M_{f(x)}$:



$$\Rightarrow L(M_{f(x)}) = \{ y \mid y \text{ ist keine akz. Berechnung von } M_x \text{ auf } \varepsilon \}$$

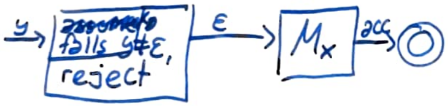
damit ist f berechenbar, berechnet von M_f.

$\square$

$$L = \{ w \in \{0,1\}^* \mid \exists x \in \{0,1\}^* \text{ mit } L(M_w) = \{x\} \}$$

Rice: Nichttriviale Eigenschaft $P(L) = \{ \exists x \mid L = \{x\} \} \Rightarrow$ nicht entscheidbar.

oder: $HP_\varepsilon \leq L$: $M_{f(x)}$:



$$L(M_{f(x)}) = \begin{cases} \{\varepsilon\} & \text{falls } \varepsilon \in L(M_x) \\ \emptyset & \text{sonst} \end{cases}$$

$\{\varepsilon\} \in L \quad \emptyset \notin L$

Also L nicht co-semi-entscheidbar.

Zählermaschine $M = \langle \Sigma, Q, C, \delta, i, F \rangle$

mit Alphabet Σ, endlich Menge Q von Kontrollzustände, endliche Menge C von Zählern,

Transitionsrelation $\delta \subseteq Q \times (\Sigma \cup (C \times \{++, --, =0\})) \times Q$,

initialem Zustand $i \in Q$, finalen Zuständen $F \subseteq Q$.

Das induzierte Transitionssystem besteht aus

Konfigurationen aus $Q \times \Sigma^* \times \mathbb{N}^C$ ($f \in \mathbb{N}^C$ heißt $f : C \to \mathbb{N}$)

Transitionen $\langle p, v, \vec{a} \rangle \to \langle q, w, \vec{b} \rangle$ falls

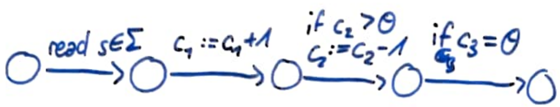
$\langle p, s, q \rangle \in \delta$, $v = s.w$, $\vec{a} = \vec{b}$, $s \in \Sigma$

$\langle p, c, ++, q \rangle \in \delta$, $v = w$, $\vec{b}_d = \begin{cases} \vec{a}_d & \text{falls } d \neq c \\ \vec{a}_d + 1 & \text{sonst} \end{cases}$

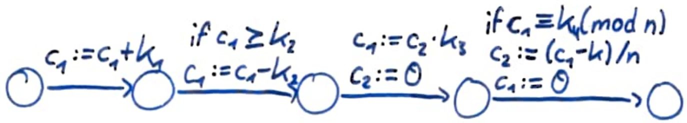
$\langle p, c, --, q \rangle \in \delta$, " " $\begin{cases} \\ -1 \end{cases}$

und $\vec{a}_c > 0$

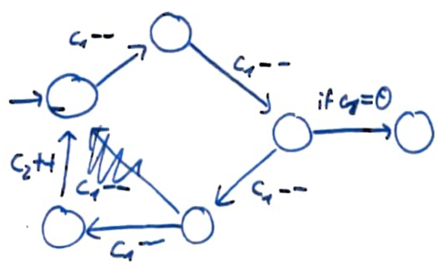
$\langle p, c=0, q \rangle \in \delta$, $v = w$, $\vec{a} = \vec{b}$, $\vec{a}_c = 0$



Es gibt Subroutinen:



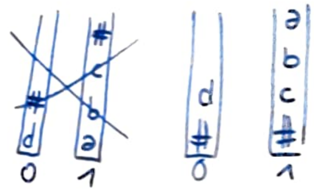
z.B. für $n=4$, $k=2$



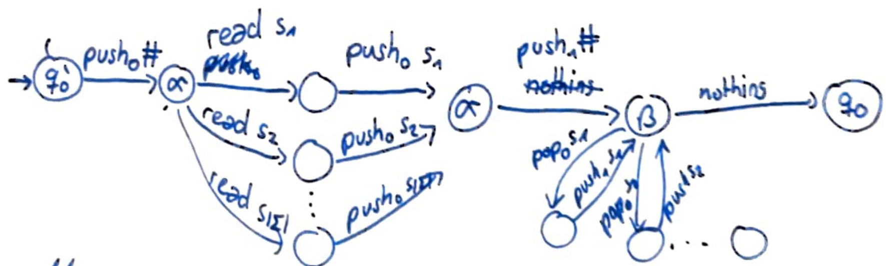
Satz: $\{L(M) \mid M\ TM\} \overset{1}{\subseteq} \{L(M) \mid M\ 2\text{-PDA}\} \overset{2}{\subseteq} \{L(M) \mid M\ 4\text{-ZM}\}$

$\underset{\text{direkt aus Def.}}{\subseteq} \{L(M) \mid M\ ZM\} \overset{3}{\subseteq} \{L(M) \mid M\ TM\}$

Idee zu 1: 

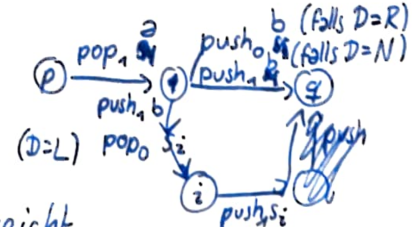
⇝ 

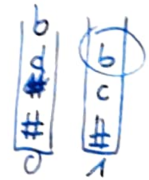
Sei M eine TM, $M = \langle Q, \Sigma, \Gamma, \sqcup, q_0, \delta, Q_F \rangle$. Dann gibt es $M' = \langle Q', \Sigma, \Gamma \cup \{\#\}, q_0', \delta', Q_F' \rangle$

M' initialisiert ~~das~~ sich: 

M' simuliert Transitionen von M:

M: $p \xrightarrow{a/b/D} q$



D=R ⇝ 

M' akzeptiert, sobald Q_F erreicht und Eingabe abgearbeitet wurde, also $Q_F' = Q_F$.

Idee zu 2: Kodiere Γ mit $\{0,1\}^{\lceil \log_2 |\Gamma| \rceil}$, speichere $\alpha \in \Gamma^*$ als $\langle \alpha \rangle$ als Binärzahl im Counter c_1. Nutze c_2, um Elemente herauszulesen (pop), einzufügen (push, mittels $c_1 := c_1 \cdot \lceil \Gamma \rceil$, $c_1 := c_1 + \langle s \rangle$)

$$\underbrace{\text{(pop)}}_{\begin{array}{l} c_1 \equiv \langle s \rangle \bmod \lceil \Gamma \rceil \\ c_2 := (c_1 - \langle s \rangle)/\lceil \Gamma \rceil \\ c_1 := 0 \\ c_1 := c_1 + c_2 \\ c_2 := 0 \end{array}}$$

$\Rightarrow$ 1-PDA $\subseteq$ 2-ZM, 2-PDA $\subseteq$ 4-ZM.