

## 1 Redukce

**Definition 1.** Necht'  $f : \Sigma^* \rightarrow \Sigma^*$  je vyčíslitelná funkce a necht'  $L_1, L_2 \subseteq \Sigma^*$  jsou jazyky. Říkáme, že  $f$  redukuje  $L_1$  na  $L_2$ , pokud pro  $w \in L_1$  právě když  $f(w) \in L_2$ . Funkci  $f$  pak říkáme redukce jazyka  $L_1$  na  $L_2$ . Pokud existuje redukce jazyka  $L_1$  na  $L_2$ , říkáme, že  $L_1$  je redukovatelný na  $L_2$ . Tento fakt zapisujeme  $L_1 \leq_m L_2$ .

**Theorem 1.** Necht'  $L_1 \leq_m L_2$ , pak platí:

- pokud  $L_2$  je rekurzivní, pak i  $L_1$  je rekurzivní.
- pokud  $L_2$  je částečně rekurzivní, pak i  $L_1$  je částečně rekurzivní.

*Důkaz.* Dokážeme pouze první tvrzení, důkaz druhého tvrzení je velmi podobný. Necht'  $R$  je TS provádějící redukci  $f$  z  $L_1$  na  $L_2$  a  $M_2$  je TS rozhodující  $L_2$ . Sestrojíme  $M_1$ , který rozhoduje  $L_1$ , TS  $M_1$  pro vstupní slovo  $w$ :

1. spustí TS  $R$  pro  $w$  a vypočítá tak  $f(w)$ .
2. spustí TS  $M_2$  pro  $f(w)$ ; pokud TS  $M_2$  přijme  $f(w)$ , TS  $M_1$  přijme  $w$ , pokud TS  $M_2$  zamítne  $f(w)$ , TS  $M_1$  zamítne  $w$ .

Pokud  $w \in L_1$ , pak  $f(w) \in L_2$  a proto  $M_2$  přijme  $f(w)$ , takže  $M_1$  přijme  $w$ . Pokud  $w \notin L_1$ , pak  $f(w) \notin L_2$  a proto  $M_2$  zamítne  $f(w)$ , takže  $M_1$  zamítne  $w$ . A tedy  $M_1$  rozhoduje  $L_1$ .

Větu 1 využijeme spíše v pozměněném tvaru:

**Theorem 2.** Necht'  $L_1 \leq_m L_2$ , pak platí:

- pokud  $L_1$  není rekurzivní, pak ani  $L_2$  není rekurzivní.
- pokud  $L_1$  není částečně rekurzivní, pak ani  $L_2$  není částečně rekurzivní.

*Důkaz.* plyne přímo z věty 1.

## 2 Příklady redukcí

**Proposition 1.**  $L_{ne} = \{[M] \mid L(M) \neq \emptyset\}$  není rekurzivní.

*Důkaz.* Tvrzení dokážeme redukcí univerzálního jazyka  $L_U$  na  $L_{ne}$  a aplikací věty 2.

Redukci reprezentujeme pomocí TS  $R$ , který ji vyčísluje:

TS  $R$  pro vstupní slovo  $[M, w]$ , kde  $M$  je TS a  $w$  je vstupní slovo nad jeho vstupní abecedou, provádí:

1. Sestaví TS  $M'$ , který pro vstupní slovo  $x$  provádí:
  - (a) Simuluje činnost TS  $M$  pro vstupní slovo  $w$ .
  - (b) Pokud  $M$  přijme  $w$ , pak  $M'$  přijme  $x$ .
  - (c) Pokud  $M$  zamítne  $w$ , pak  $M'$  zamítne  $x$ .
2. Zapiše na pásku  $[M']$  a zastaví.

TS  $R$  tedy provádí konverzi  $[M, w]$  na  $[M']$  tak, že platí:

- Pokud  $[M, w] \in L$ , tj.  $M$  nepřijímá  $w$ , pak  $M'$  přijímá jakékoli vstupní slovo  $x$  a tedy přijímá neprázdný jazyk, takže  $[M'] \in L_{\neq}$ .
- Pokud  $[M, w] \notin L$ , tj.  $M$  pro  $w$  cyklí nebo jej zamítá, pak  $M'$  buďto cyklí při simulaci  $M$  pro  $w$ , nebo zamítá  $x$ , protože  $M$  zamítlo  $w$ . Jakékoli vstupní slovo  $x$  je tedy zamítnuto a  $L(M) = \emptyset$ , takže  $[M'] \in L_{\neq}$ .

Tím jsme ukázali, že  $L_U \leq_m L_{ne}$ , a z věty 2 dostáváme, že  $L_{ne}$  není rekurzivní.

### 3 Riceova věta

#### Theorem 3. Riceova věta:

Nechť  $P$  je problém týkající se  $TS$ , který splňuje:

- Pokud  $L(M_1) = L(M_2)$ , pak  $[M_1] \in P$  p.k.  $[M_2] \in P$ ,
- Existuje  $[M] \in P$ , existuje  $[M'] \notin P$ .

$P$  je neřešitelný.

Důkaz. Uvažujeme, že:

- $\emptyset \notin P$  ... myšleno tak, že žádný  $\langle M \rangle$  t.ž.  $L(M) = \emptyset$  není v  $P$ .
- $L \in P$  ... tzn, že každý  $[M_L]$  t.ž.  $L(M_L) = L$  je v  $P$ .

TS  $R$  pro  $[M, w]$ :

- sestrojí  $M'$  t.ž. TS  $M'$  pro  $x$ :
  - simuluje  $M$  pro  $w$ , pokud přijme
  - simuluje  $M_L$  pro  $x$ , pokud přijme TS  $M'$  přijme.

Pokud  $M$  přijme  $w$ ,  $L(M') = L$ , a tedy  $[M'] \in P$ .

Pokud  $M$  nepřijme  $w$ ,  $L(M') = \emptyset$ , a tedy  $[M'] \notin P$ .

$L_U$  je neřešitelný, tedy  $P$  je neřešitelný.

To jsme uvažovali  $\emptyset \notin P$ . Pro případ  $\emptyset \in P$ , to můžeme dokázat stejně pro  $\overline{P}$ .

### 4 Postův problém přiřazení

Mějme množinu uspořádaných dvojic řetězců jako je například tato:

$$\left\{ \left[ \begin{array}{c} 0 \\ 100 \end{array} \right], \left[ \begin{array}{c} 01 \\ 00 \end{array} \right], \left[ \begin{array}{c} 110 \\ 11 \end{array} \right] \right\} \quad (1)$$

Řešením v sadě je sekvence dominových kostek, tak že konkaténace řetězců v horní části je rovna konkaténaci řetězců v dolní části. Dominové kostky se v této sekvenci mohou opakovat. Například řešení v sadě 1 je

$$\left[ \begin{array}{c} 110 \\ 11 \end{array} \right], \left[ \begin{array}{c} 01 \\ 00 \end{array} \right], \left[ \begin{array}{c} 110 \\ 11 \end{array} \right], \left[ \begin{array}{c} 0 \\ 100 \end{array} \right],$$

protože konkaténací řetězců v horní i dolní části dostaneme stejný řetězec 110011100.

Postův problém přiřazení je pak definován takto:

**Definition 2.** Postův problém přiřazení je rozhodovací problém „Existuje v sadě  $S$  řešení?“

Postův problém přiřazení s inicializací je rozhodovací problém „Existuje v sadě  $S$  řešení začínající danou kostkou?“

Vyjádřeny jako jazyky:

- Postův problém přiřazení:

$$PPP = \{[S] \mid S \text{ je sada, která má řešení}\}.$$

- Postův problém přiřazení s inicializací:

$$PPP_{init} = \{[S, \frac{a}{b}] \mid S \text{ je sada, která má řešení začínající kostkou } \frac{a}{b}\}.$$

**Theorem 4.** PPP pro  $|\Sigma| \geq 2$  není rekurzivní a  $PPP_{init}$  pro  $|\Sigma| \geq 2$  není rekurzivní.

Důkaz. Ukážeme, že  $L_U \leq_m PPP_{init}$ : Potřebujeme tedy sestavit TS  $R$ , který provádí konverzi  $[M, w]$  na  $[S, \frac{a}{b}]$ , a to tak, že  $M$  přijme  $w$ , právě když  $S$  má řešení začínající kostkou  $\left[ \frac{a}{b} \right]$ .

TS  $R$  pro  $[M, w]$ :

1. Sestaví stroj TS  $T = \langle Q, \Sigma, \Gamma, \delta, q_{start}, q_+, q_- \rangle$ , který je ekvivalentní s  $M$ , ale nikdy se nepokusí přejít levý okraj pásky.
2. Ke stroji  $T$  sestaví následující sadu  $S$ :

(a) Vloží  $\left[ \frac{\#}{\#q_{\text{start}}w_1w_2 \dots w_n\#} \right]$  so  $S$ , kde  $w_1w_2 \dots w_n = w$ .

(b) Pro každé  $a, b \in \Gamma$  a každé  $q, r \in Q$ , kde  $q \neq q_-$ ,

pokud  $\delta(q, a) = (r, b, R)$ , vloží  $\left[ \frac{qa}{br} \right]$  do  $S$

(c) Pro každé  $a, b \in \Gamma$  a každé  $q, r \in Q$ , kde  $q \neq q_-$ ,

pokud  $\delta(q, a) = (r, b, L)$ , vloží  $\left[ \frac{cqa}{cbr} \right]$  do  $S$

(d) Pro každé  $a \in \Gamma$ ,

vloží  $\left[ \frac{a}{a} \right]$  do  $S$

(e) Vloží  $\left[ \frac{\#}{\#} \right]$  a  $\left[ \frac{\#}{\#} \right]$  do  $S$ .

(f) Pro každé  $a \in \Gamma$

vloží  $\left[ \frac{aq_+}{q_+} \right]$  a  $\left[ \frac{q_+a}{q_+} \right]$  do  $S$

(g) Vloží  $\left[ \frac{q_+\#\#}{\#} \right]$  do  $S$

3. zapíše na pásku  $[S, \left[ \frac{\#}{\#q_{\text{start}}w_1w_2 \dots w_n\#} \right]]$  a skončí

Pokud  $M$  přijímá  $w$  – tedy  $T$  přijímá  $w$ , existuje přijímající výpočet výpočet TS  $T$  nad  $w$ . Sestrojíme sadu  $S$ , ve které existuje jen řešení odpovídající tomuto výpočtu. Toto řešení bude mít v horní i dolní části celý výpočet TS  $T$  nad  $w$ .

Dále, ukážeme že  $\text{PPP}_{\text{init}} \leq_m \text{PPP}$ : Potřebujeme tedy konvertovat  $[S, \left[ \frac{a}{b} \right]]$  na  $[S']$ , tak že sada  $S$  má řešení začínající  $\left[ \frac{a}{b} \right]$ , právě když  $S'$  má řešení.

Pro řetězec  $x$  označme  $(x_{\#})$  řetězec, který vznikne z  $x$  tak, že se přidá symbol  $\#$  za každý jeho symbol, a označme  $(_{\#}x)$  řetězec, který vznikne z  $x$  tak, že se přidá symbol  $\#$  před každý jeho symbol. Například  $(001_{\#}) = 0\#0\#1\#$  a  $(_{\#}001) = \#0\#0\#1$ .

TS  $R$  pro vstupní slovo  $[S, \left[ \frac{a}{b} \right]]$ :

1. Sestaví  $S'$  tak, že pro každou kostku  $\left[ \frac{x}{y} \right] \in S$  dáme  $\frac{(x_{\#})}{(_{\#}y)}$  do  $S'$ .

2. Přidá  $\left[ \frac{\square}{\#\square} \right]$  do  $S'$ .

3. Přidá  $\left[ \frac{\#(a_{\#})}{(_{\#}b)} \right]$  do  $S'$ .

4. Zapíše  $[S']$  a skončí.

Tedy pro každou kostku z  $S$  TS  $R$  vytvoří kostku tak, že v horním řetězci umístíme znak  $\#$  před každý jeho symbol a v dolním řetězci umístíme znak  $\#$  za každý jeho symbol. Dále se přidá  $\left[ \frac{\square}{\#\square} \right]$  do  $S'$ , která má soužit jako „ukončovací“ kostka.

V tomto okamžiku  $S'$  nemá žádné řešení – chybí kostka, kterou by se dalo začít. V bodě 3 přidáváme kostku, kterou je možné začít a která odpovídá iniciální kostce  $\left[ \frac{a}{b} \right]$ . Jakékoli řešení sady  $S'$  tedy musí začínat touto kostkou. Takové řešení existuje

právě tehdy, když sada  $S$  má řešení začínající  $\left[ \frac{a}{b} \right]$ .