Úkol 2

1. příklad

Uvažujte jazyk $L = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$, kde $\#_x(w)$ značí počet výskytů symbolů x v řetězci w. Dokažte, že jazyk L je bezkontextový. Postupujte následovně:

(a) Nejdříve navrhněte gramatiku G, která bude mít za cíl jazyk L generovat.

(b) Poté pomocí indukce k délce slova $w \in L$ dokažte, že $L = L(G)$.

Řešení:

(a) Nechť $\boldsymbol{G}$ je následující gramatika generující jazyk L:
$\boldsymbol{G} = (\{S\}, \{a, b\}, \{S \to aSbS \mid bSaS \mid \varepsilon\}, S)$

(b) Důkaz, že $\boldsymbol{L = L(G)}$ provedeme dokázáním, že i) $\boldsymbol{L(G) \subseteq L}$ a ii) $\boldsymbol{L \subseteq L(G)}$ matematickou indukcí k délce slova $w \in L$.

i) Důkaz, že $\boldsymbol{L(G) \subseteq L}$.

a) Důkaz vzhledem k délce slova $i = 0$.

- Slovo o délce 0, $\varepsilon : |\varepsilon| = 0$ lze vygenerovat pravidlem $S \to \varepsilon$ gramatiky G, tj. $\varepsilon \in L(G)$.
- Zároveň platí, že $\varepsilon \in L$.
- Pro $i = 0$ dokazované tvrzení tedy platí.

b) Předpokládejme, že pro slovo w platí, že $|w| \leq i \land S \Rightarrow^* w : w \in L$. Na základě tohoto indukčního předpokladu ukažme, že dokazované tvrzení platí i pro slova délky $i + 2$.

- Gramatika G derivuje následující řetězce $S \Rightarrow aSbS \Rightarrow^* aw'bw'' = w_1$, $S \Rightarrow bSaS \Rightarrow^* bw'aw'' = w_2$.
- Dle indukčního předpokladu platí, že když $S \Rightarrow^* w' \land S \Rightarrow^* w''$, tak $w' \in L \land w'' \in L$, protože $w' \in L(G) \land w'' \in L(G) \land |w'| \leq i \land |w''| \leq i$, a tedy $w', w'' \in \{a, b\}^* : \#_a(w') = \#_b(w') \land \#_a(w'') = \#_b(w'')$.
- Z výše uvedeného plyne, že $\#_a(w_1) = \#_a(w') + \#_a(w'') + 1 = \#_b(w') + \#_b(w'') + 1 = \#_b(w_1) \Rightarrow w_1 \in L$.
- A zároveň platí, že $\#_a(w_2) = \#_a(w') + \#_a(w'') + 1 = \#_b(w') + \#_b(w'') + 1 = \#_b(w_2) \Rightarrow w_2 \in L$.

c) $\boldsymbol{L(G) \subseteq L}$ **tedy platí.**

ii) Důkaz, že $\boldsymbol{L \subseteq L(G)}$.

a) Důkaz vzhledem k délce slova $i = 0$.

- Slovo o délce 0, $\varepsilon : |\varepsilon| = 0 \land \varepsilon \in L$.
- Zároveň ε lze vygenerovat gramatikou G použitím pravidla $S \to \varepsilon$, tj. $\varepsilon \in L(G)$.
- Pro $i = 0$ tedy dokazované tvrzení platí.

b) Předpokládejme, že pro slovo w platí, že $|w| \leq i \land w \in L : S \Rightarrow^* w$. Na základě tohoto indukčního předpokladu ukážeme, že dokazované tvrzení platí i pro slova délky $i + 2$.

- Pro řetězce z množiny $W = \{w \in L \mid \#_a(w) = \frac{i+2}{2} \land \#_b(w) = \frac{i+2}{2} \land |w| = i + 2\}$ platí, že $\forall\, w \in W : \exists\, w' \in L : \#_a(w) = \#_a(w') + 1 = \#_b(w') + 1 = \#_b(w) \land |w'| \leq i$.

- Dle indukčního předpokladu tedy platí, že $\forall\, w \in W : \exists\, w' \in L : S \rightarrow w'$.
- Řetězce z množiny W lze generovat gramatikou G následovně:

  $S \Rightarrow aSbS \Rightarrow aSb \Rightarrow aw'b$

  $S \Rightarrow aSbS \Rightarrow abS \Rightarrow abw'$

  $S \Rightarrow bSaS \Rightarrow bSa \Rightarrow bw'a$

  $S \Rightarrow bSaS \Rightarrow baS \Rightarrow baw'$
- Všechny výše uvedené řetězce patří do jazyka L, protože $w' \in L$, jak bylo ukázáno výše.

c) $\boldsymbol{L \subseteq L(G)}$ **tedy platí.**

□

2. příklad

Uvažujte *doprava čtený jazyk* TS M, značený jako $L^P(M)$, který je definován jako množina řetězců, které M přijme v běhu, při kterém nikdy nepohne hlavou *doleva* a nikdy nepřepíše žádný symbol na pásce za jiný. Dokažte, zda je problém prázdnosti doprava čteného jazyka TS M, tj. zda $L^P(M) = \emptyset$, rozhodnutelný:

- pokud *ano*, napište algoritmus v pseudokódu, který daný problém bude rozhodovat;
- pokud *ne*, dokažte nerozhodnutelnost redukcí z jazyka HP.

Řešení:

Problém prázdnosti doprava čteného jazyka Turingova stroje M, tj. $L^P(M) = \emptyset$, **je rozhodnutelný**. Důkazem nechť je následující algoritmus, který tento problém rozhoduje.

Algoritmus:

Vstup: Deterministický Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$ s přechodovou parciální funkcí δ definovanou následovně: $(Q \setminus \{q_f\}) \times \Gamma \rightarrow Q \times (\Gamma \cup \{L, R\})$, kde $L, R \notin \Gamma$.

Výstup: $\begin{cases} \texttt{TRUE} & \text{pokud } L^P(M) = \emptyset \\ \texttt{FALSE} & \text{jinak, tj. pokud } L^P(M) \neq \emptyset \end{cases}$

Metoda:

1. Nechť $R_\delta \subseteq Q \times Q$ je binární relace, která popisuje, zda je v Turingově stroji M možný přímý přechod (definovaný jazykem $L^P(M)$) mezi danou dvojicí stavů (p, q), definována na základě přechodové funkce δ následovně:
   $R_\delta = \{(p, q) \in Q \times Q \mid \exists\, \gamma_n \in \Gamma : ((q, R) \in (p, \gamma_n)) \vee ((q, \gamma_n) \in (p, \gamma_n))\}$
2. Nechť R_δ^+ je tranzitivní uzávěr relace R_δ vypočtený Warshallovým algoritmem.
3. Nechť výstup algoritmu já dán predikátem φ definovaným následovně:
   $\varphi : \neg\, (\exists\, p, q \in Q : p = q_0 \wedge q = q_f \wedge (p, q) \in R_\delta^+)$

□

3. příklad

Uvažujte jazyk $L_{42} = \{\langle M \rangle \mid$ TS M zastaví na některém vstupu tak, že páska bude obsahovat právě 42 neblankových symbolů$\}$. Dokažte pomocí redukce, že L_{42} je nerozhodnutelný. Uveďte ideu důkazu částečné rozhodnutelnosti L_{42}.

Řešení:

Důkaz, že jazyk L_{42} je nerozhodnutelný.
Provedeme důkaz redukcí z problému zastavení Turingova stroje (HP).

- Jazyk, který charakterizuje HP bude vypadat následovně:
  $HP = \{\langle M \rangle \# \langle w \rangle \mid M$ je Turingův stroj takový, že na řetězci w zastaví$\}$, kde $\langle M \rangle$ je kód Turingova stroje M a $\langle w \rangle$ je kód řetězce w.
- Zadaný jazyk $L_{42} = \{\langle M \rangle \mid M$ je Turingův stroj takový, že zastaví na některém vstupu tak, že páska bude obsahovat právě 42 ne-blankových symbolů$\}$, kde $\langle M \rangle$ je kód Turingova stroje M, bude charakterizovat problém, který tento jazyk reprezentuje.
- Navrhneme redukci $\sigma : \{0, 1, \#\}^* \to \{0, 1\}^*$ z jazyka HP na jazyk L_{42}.
- Redukce σ přiřadí řetězci $x \in \{0, 1, \#\}^*$ řetězec $\langle M_x \rangle$, což je kód Turingova stroje M_x, který pracuje následovně:
  1. M_x smaže svůj vstup w.
  2. M_x zapíše na vstupní pásku řetězec x, který má uložen v konečném stavovém řízení.
  3. M_x ověří, zda x má strukturu $x_1 \# x_2$, kde x_1 je kód Turingova stroje a x_2 je kód jeho vstupu. Pokud ne, odmítne.
  4. M_x odsimuluje na řetězci s kódem x_2 běh Turingova stroje s kódem x_1. Pokud simulace skončí, smaže svou pásku, zapíše na ni 42 libovolných ne-blankových symbolů a následně přijme. Jinak cyklí.
- Redukci σ je možné implementovat úplným Turingovým strojem M_σ, který pro vstup x vyprodukuje kód Turingova stroje M_x. Tento sestává z následujících komponent:
  1. Komponenta, která maže vstupní pásku — lze předpřipravit a pak M_σ jen vypíše patřičný kód.
  2. M_σ vypíše kód Turingova stroje, který jen zapíše na vstup řetězec $a_1, a_2, \ldots, a_n$ — opakovaný sekvenční zápis a posuv doprava.
  3. M_σ vypíše kód Turingova stroje, který na vstupu ověří, zda se jedná o platnou instanci HP a pokud ne, odmítne. (Test na členství v regulárním jazyce.)
  4. M_σ vypíše kód Turingova stroje, který spustí univerzální Turingův stroj na Turingův stroj s kódem x_1 a vstupem s kódem x_2.
- M_σ zajistí sekvenční předávání řízení mezi jednotlivými komponentami.
- Nyní zkoumejme jazyk Turingova stroje M_x:
  a) $L(M_x) = \emptyset \Leftrightarrow$ (x nemá strukturu $x_1 \# x_2$ pro kód Turingova stroje x_1 a kód vstupu x_2) $\vee$ (x má strukturu $x_1 \# x_2$, kde x_1 je kód Turingova stroje a x_2 kód vstupu, ale Turingův stroj s kódem x_1 na vstupu s kódem x_2 nezastaví).
  b) $L(M_x) = \Sigma^* \Leftrightarrow$ (x má strukturu $x_1 \# x_2$, kde x_1 je kód Turingova stroje a x_2 je kód vstupu a Turingův stroj s kódem x_1 zastaví na vstupu s kódem x_2) $\wedge$ (M_x má po přijetí na pásce právě 42 ne-blankových symbolů).
- Konečně ukážeme, že redukce σ zachová členství v jazyce:
  $\forall\ x \in \{0, 1, \#\}^* : (\sigma(x) = \langle M_x \rangle \in L_{42}) \Leftrightarrow (L(M_x) = \Sigma^*) \Leftrightarrow$ ((x má strukturu $x_1 \# x_2$, kde x_1 je kód Turingova stroje a x_2 kód vstupu) $\wedge$ (Turingův stroj s kódem x_1 zastaví na vstupu s kódem x_2) $\wedge$ (M_x má po přijetí na pásce právě 42 ne-blankových symbolů)) $\Leftrightarrow x \in HP$.

$\square$

Idea důkazu, že jazyk L_{42} je částečně rozhodnutelný.
Lze sestavit Turingův stroj M' rozhodující jazyk L_{42} následujícím způsobem:

- M' zkontroluje, zda na vstupu má platný kód Turingova stroje M. Pokud ne, odmítne.
- M' na pomocné pásce postupně simuluje běh Turingova stroje M na jednotlivých vstupních řetězcích w. Jednotlivé páskové konfigurace jsou na pomocné pásce vhodně uspořádány.
- M' vždy projde všechny rozpracované simulace a na každé dále simuluje jeden krok výpočtu.
- Pokud byl v některém kroce řetězec přijat, M' taky přijme. V opačném případě se přidá pásková konfigurace pro další řetězec a kroky simulace se opakují.

□

4. příklad

Uvažujte programovací jazyk **`Karel@TIN`** se zadanou gramatikou a sémantikou.
Dokažte, že programovací jazyk **`Karel@TIN`** je Turingovksy úplný, tj., dokažte, že

(a) pro každý TS M nad abecedou $\{0,1\}$ a řetězec $w \in \{0,1\}^*$ lze sestrojit program P_M v jazyce **`Karel@TIN`** a zvolit počáteční konfiguraci prostředí C_M tak, že P_M skončí s návratovou hodnotou 1 právě tehdy, když $w \in L(M)$;

(b) pro každý program P v jazyce **`Karel@TIN`** a počáteční konfiguraci C lze spustit TS M_P a řetězec $w \in \{0,1\}^*$ tak, že $w \in L(M_P)$ právě tehdy, když robot Karel po interpretaci programu P z počáteční konfigurace C skončí s návratovou hodnotu 1.

Řešení:

(a) Počáteční konfigurace $C_M = ((0,0), \uparrow, g)$.
Kódování symbolů: $0 \simeq 0$, $1 \simeq 1$, $\Delta \simeq 2$.

`write_blank`:

```
0 if empty: goto 3;
1 lift-screw;
2 if not empty: goto 1;
3 drop-screw;
4 drop-screw;
```

`write_0`:

```
0 if empty: goto 3;
1 lift-screw;
2 if not empty: goto 1;
3 turn left;
4 turn left;
5 turn left;
6 turn left;
```

`write_1`:

```
if empty: goto 3;
lift-screw;
if not empty: goto 1;
drop-screw;
```

`move_forward`:

```
step;
```

`move_backward`:

```
turn left;
turn left;
step;
turn left;
turn left;
```