Úkol č. 3 do předmětu Teoretická informatika

Vojtěch Havlena (xhavle03)

1. Příklad

Jazyk Turingova stroje na obrázku lze pomocí regulárního výrazu zapsat jako $a(baa)^*$. Gramatika G generující tento jazyk je potom definována následovně $G = (\{S, B\}, \{a, b\}, P, S)$, kde množina přepisovacích pravidel je dána jako

$$P = \{S \to aB, B \to baaB, B \to \varepsilon\}.$$

Přijímající běh Turingova stroje pro řetězec $abaa$.

$$\begin{aligned}
(0, \Delta abaa\Delta^\omega, 0) &\vdash (1, \Delta abaa\Delta^\omega, 1) \vdash (2, \Delta abaa\Delta^\omega, 2) \vdash (3, \Delta abaa\Delta^\omega, 3) \\
&\vdash (1, \Delta abaa\Delta^\omega, 4) \vdash (2, \Delta abaa\Delta^\omega, 5) \vdash (5, \Delta abaa\Delta^\omega, 4) \\
&\vdash (6, \Delta aba\Delta^\omega, 4) \vdash (5, \Delta aba\Delta^\omega, 3) \vdash (6, \Delta ab\Delta^\omega, 3) \\
&\vdash (5, \Delta ab\Delta^\omega, 2) \vdash (6, \Delta a\Delta^\omega, 2) \vdash (5, \Delta a\Delta^\omega, 1) \\
&\vdash (6, \Delta^\omega, 1) \vdash (5, \Delta^\omega, 0) \vdash (7, \Delta^\omega, 1) \\
&\vdash (8, \Delta Y\Delta^\omega, 1) \vdash (9, \Delta Y\Delta^\omega, 0)
\end{aligned}$$

Derivace řetězce $abaa$ gramatikou G.

$$S \Rightarrow aB \Rightarrow abaaB \Rightarrow abaa$$

2. Příklad

Důkaz toho, že problém prázdnosti jazyka daného Turingova stroje (EMP) není ani částečně rozhodnutelný provedu redukcí z co-HP, kde

$$\text{EMP} = \{\langle M\rangle \mid L(M) = \emptyset\} \text{ a co-HP } = \{\langle M\rangle\#\langle w\rangle \mid M \text{ nezastaví při } w\}.$$

Jazyk co-HP není ani rekurzivně vyčíslitelný jazyk. Navrhneme tedy totální funkci $\sigma : \{0, 1, \#\}^* \to \{0, 1\}^*$, která zachovává členství v jazycích co-HP a EMP a je implementovatelná úplným TS. Funkce σ pro zadaný vstup $x \in \{0, 1\}^*$ vygeneruje kód TS M_x (vygeneruje tedy $\langle M_x\rangle$) a M_x pracuje následovně:

(a) M_x smaže obsah své vstupní pásky.

(b) M_x zapíše na svou pásku řetězec x.

(c) M_x otestuje, zda řetězec x má strukturu $x_1\#x_2$, kde $x_1 = \langle M\rangle$ (x_1 je tedy kód TS M) a x_2 je kód jeho vstupu w. Pokud takovou strukturu nemá, přijme.

(d) Jinak M_x s využitím univerzálního TS odsimuluje běh TS M na vstupu w. Pokud M na w zastaví, M_x přijme řetězec x (jinak tedy cyklí).

Redukci σ lze implementovat úplným TS. Stačí, aby tento úplný TS byl schopen vygenerovat TS, který předává řízení mezi 4 komponentami, které implementují body (a) – (d). Komponenty (a), (b), (d) jsou nezávislé na x. Pro vypsání x na vstupní pásku stačí vygenerovat kód TS, který postupně zapisuje symboly z x na vstupní pásku a posunuje hlavu o jednu pozici doprava.

Jazyk $L(M_x)$ je buď prázdný nebo $\{0,1\}^*$, přičemž

(a) $L(M_x) = \emptyset \Leftrightarrow x = x_1\#x_2$, kde $x_1 = \langle M \rangle$ (x_1 je kód TS M) a $x_2 = \langle w \rangle$ (kód jeho vstupu) a M na w nezastaví.

(b) $L(M_x) = \{0,1\}^* \Leftrightarrow x$ není správně zformovaná instance co-HP nebo $x = x_1\#x_2$, kde $x_1 = \langle M \rangle$ (x_1 je kód TS M) a $x_2 = \langle w \rangle$ (kód jeho vstupu) a M na w zastaví.

V posledním kroku důkazu zbývá ukázat, že σ zachovává členství v jazyce, tj. $x \in$ co-HP $\Leftrightarrow \sigma(x) \in$ EMP. Tedy

$\sigma(x) = \langle M_x \rangle \in \text{EMP} \Leftrightarrow L(M_x) = \emptyset \Leftrightarrow x = x_1\#x_2$, kde $x_1 = \langle M \rangle$ (x_1 je kód TS M) a $x_2 = \langle w \rangle$ (kód jeho vstupu) a M na w nezastaví $\Leftrightarrow x \in$ co-HP. Problém prázdnosti jazyka daného Turingova stroje tedy není ani částečně rozhodnutelný.

3. Příklad

(a) Předpokládejme, že existuje konečná množina P řetězců, pro kterou je HP P-rozhodnutelný. Tedy existuje TS M_P, který P-rozhoduje HP. Potom je jazyk $PHP = HP \setminus P$ rekurzivní – úplný TS M, který rozhoduje jazyk PHP by pracoval následovně: TS M by zkontroloval, zda vstupní řetězec je obsažen v množině P (množina P je konečná, stačí projít všechny prvky a porovnávat). Pokud ano, M zamítne. Jinak pomocí univerzálního TS simuluje běh TS M_P, který P-rozhoduje HP se vstupním řetězcem. Pokud M_P přijme, tak přijme i M. Pokud M_P zamítne, zamítne i M.

Dále je možné si jazyk HP vyjádřit následovně: $HP = PHP \cup (HP \cap P)$. Vzhledem k tomu, že P je konečná množina, je i $HP \cap P$ konečná množina a tedy i rekurzivní jazyk. Vzhledem k tomu, že PHP je rekurzivní jazyk a rekurzivní jazyky jsou uzavřeny vůči sjednocení, je i $HP = PHP \cup (HP \cap P)$ rekurzivní jazyk, což je spor. Konečná množina P řetězců, pro kterou je HP P-rozhodnutelný tedy neexistuje.

(b) Za P si zvolím celou množinu HP, tedy $P = HP$. Tato množina je nekonečná. Dále uvažujme TS M takový, který pro každý vstup okamžitě zastaví a nepřijme. Tedy $L(M) = \emptyset$. Tento TS pro každý vstup zastaví a ze slov mimo P přijímá právě ta, která patří do HP (což nejsou žádná slova). A tedy TS M P-rozhoduje HP.

(c) Předpokládejme, že tvrzení platí, tedy pro všechny nekonečné množiny řetězců P je HP P-rozhodnutelný. Když platí pro všechny nekonečné, tak platí i pro

$P = \{0,1\}^*$. Nechť pro $x \in \{0,1\}^*$ je M_x TS s kódem x, je-li x validní kód TS. Jinak M_x je pevně zvolený TS, který pro libovolný vstup okamžitě zastaví. Posloupnost $M_\varepsilon, M_0, M_1, \ldots$, idexovaná řetězci z $\{0,1\}^*$ zahrnuje všechny Turingovy stroje nad $\{0,1\}^*$. Podle předpokladu existuje TS K, který P-rozhoduje HP. Tento TS pro vstup $\langle M\rangle\#\langle w\rangle$ (platí, že $\langle M\rangle\#\langle w\rangle \notin P$):

- Zastaví a přijme právě tehdy, když TS M zastaví na w.
- Zastaví a odmítne právě tehdy, když TS M cyklí na w.

Dále je možné sestavit TS N, který pro vstup $x \in \{0,1\}^*$:

- Sestaví M_x z x a zapíše $\langle M_x\rangle\#\langle x\rangle$ na svou pásku.
- Dále simuluje TS K na vstup $\langle M_x\rangle\#\langle x\rangle$ a přijme, pokud K odmítne. Pokud K přijme, přejde do nekonečného cyklu.

Potom ale N zastaví na $x \Leftrightarrow K$ odmítne $\langle M_x\rangle\#\langle x\rangle \Leftrightarrow M_x$ cyklí na x. TS N se tedy liší od každého M_x alespoň na jednom řetězci. Což je ovšem spor s tím, že posloupnost $M_\varepsilon, M_0, M_1, \ldots$ zahrnuje všechny Turingovy stroje nad $\{0,1\}$. Tvrzení tedy neplatí.

4. **Příklad** Pro simulaci Turingova stroje (TS) pomocí rendez-vous sítí (RVS) je nejprve nutné uvažovat kódování konfigurace TS pomocí rendez-vous sítí. Neformálně každý proces v RVS představuje políčko na pásce TS. Tyto procesy jsou lineárně zřetězeny pomocí komunikačních kanálů. Každý proces (vyjma procesu představující 1. políčko pásky a poslední neblankové políčko pásky) je pomocí jednoho kom. kanálu spojen s jedním sousedem a pomocí druhého kom. kanálu s druhým sousedem (lze si představit jako jakýsi lineárně vázaný seznam).

   Ve stavech automatu popisující chování procesů ukládám informaci o aktuálním stavu TS, páskovém symbolu, který je uložen na políčku pásky reprezentovaném procesem, číslo procesu pro oddělení jednotlivých procesů a příznak, který udává, zda tento proces má řízení (= hlava TS je právě nad tímto políčkem pásky). Příznakem řízení může být označen pouze jeden proces Pro číselné oddělení jednotlivých procesů používám čísla ze $\mathbb{Z}_3$, navíc s číslem 4 pro číslo nově přidaného procesu (=posunutí hlavy TS na pozici pásky, pro kterou zatím neexistuje proces). Stav konečného automatu je tedy čtveřice.

   Abeceda je trojice stav TS (pro předávání řízení mezi procesy), identifikace procesu a příznak popisující, zda se jedná a nově vytvořený proces.

   Pro daný TS $M = (Q_T, \Sigma_T, \Gamma, \delta_T, q_0^T, q_F^T)$ je konečný automat A popisující chování procesů definován následovně: $A = (Q, \Sigma \times \{0,1\}, \delta, q_0, F)$, kde

   - $Q = Q_T \times \Gamma \times M \times \{0,1\}$, kde $M = \mathbb{Z}_3 \cup \{4\}$
   - $\Sigma = Q_T \times M \times \{0,1\}$
   - $q_0 = (q_0^T, \Delta, 4, 0)$
   - $F = \{q_F^T\} \times \Gamma \times M \times \{0,1\}$

– Přechodová funkce $\delta : Q \times \Sigma \times \{0,1\} \rightarrow 2^Q$ je dána následovně ($\oplus$ značí operaci sčítání v $\mathbb{Z}_3$ a $\ominus$ odečítání v $\mathbb{Z}_3$):

– $\forall q_1, q_2, q_3 \in Q_T, \forall a_1, a_3 \in \Gamma, \forall v_1, v_2, v_3 \in M, \forall Y_1, Y_3, new, i \in \{0,1\}$:
$$(q_3, a_3, v_3, Y_3) \in \delta((q_1, a_1, v_1, Y_1), ((q_2, v_2, new), i)) \Leftrightarrow$$
$$(new = 0 \land Y_1 = 0 \land Y_3 = 1 \land v_1 = v_2 = v_3 \neq 4 \land a_3 = a_1 \land q_3 = q_2) \lor$$
$$(new = 0 \land Y_1 = 1 \land Y_3 = 0 \land v_2 = v_1 \oplus 1 \land v_3 = v_1 \land a_3 = a_1 \land q_3 = q_2 \land \delta_T(q_1, a_1) = (q_2, R)) \lor$$
$$(new = 0 \land Y_1 = 1 \land Y_3 = 0 \land v_2 = v_1 \ominus 1 \land v_3 = v_1 \land a_3 = a_1 \land q_3 = q_2 \land \delta_T(q_1, a_1) = (q_2, L)) \lor$$
$$(new = 0 \land Y_1 = 0 \land Y_3 = 0 \land v_2 = 4 \land v_1 = v_3 \neq 4 \land a_3 = a_1 \land q_1 = q_3) \lor$$
$$(new = 0 \land Y_1 = 1 \land Y_3 = 1 \land v_2 = 4 \land v_1 = v_3 \neq 4 \land \delta_T(q_1, a_1) = (q_3, a_3)) \lor$$
$$(new = 1 \land Y_1 = 1 \land Y_3 = 0 \land v_2 = v_1 = v_3 \land a_3 = a_1 \land q_3 = q_2 \land \delta_T(q_1, a_1) = (q_2, R)) \lor$$
$$(new = 1 \land Y_1 = 0 \land Y_3 = 1 \land v_1 = 4 \land v_3 = v_2 \oplus 1 \land a_3 = a_1 \land q_3 = q_2)$$

Sémantika: q je stav TS, a je symbol na pásce, v je identifikace procesu, Y je příznak řízení, new – zda se jedná o nový proces. První tři pravidla představují předání řízení z aktivního procesu do sousedního procesu (simulace posunu hlavy TS vlevo/vpravo). Další dvě pravidla představují simulaci pravidla, kdy v TS dojde k přepsání symbolu pod hlavou (samotná hlava není posunuta). Poslední dvě pravidla simulují posun hlavy vpravo za poslední neblankový symbol na pásce (dojde k vytvoření nového procesu, tomuto procesu se přiřadí identifikační číslo a předá se mu řízení).

Způsob kódování konfigurace TS pomocí RVS. Nechť $(q, \gamma\Delta^\omega, n)$ je konfigurace TS M, kde $\gamma \in \Gamma^*$ a $|\gamma| > n$. Potom této konfiguraci odpovídá konfigurace RVS $(S, stav)$, kde $S = (A, P, K)$. Konečný automat A je pro TS M definován, jak je popsáno výše. Množina procesů $P = \{p_0, \ldots, p_{m-1}\}$, kde $m = |\gamma|$. Množina komunikačních kanálů $K = \{(p_i, 1, p_{i+1}) | p_i, p_{i+1} \in P, i \text{ je sudé}\} \cup \{(p_i, 2, p_{i+1}) | p_i, p_{i+1} \in P, i \text{ je liché}\}$. Funkce $stav$ je potom definována následovně:

$$stav(p_i) = \begin{cases} (q_0^T, \gamma_i, i \mod 3, 0) & \text{pokud } i \neq n \\ (q, \gamma_i, i \mod 3, 1) & \text{pokud } i = n \end{cases}$$

V případě, kdy máme zadanou konfiguraci RVS, tak odpovídající konfiguraci TS získáme tam, že nelezneme proces, který má nastaven příznak řízení. Z toho procesu získáme aktuální stav TS. Pozici hlavy získáme průchodem z aktivního procesu přes komunikační kanály směrem k procesům s nižším identifikačním číslem.

Tvrzení 1.1. *Nechť M je Turingův stroj a $K_1 = (q, \gamma\Delta^\omega, n)$ a $K_2 = (q', \gamma'\Delta^\omega, m)$ jsou konfigurace M, přičemž $K_1 \neq K_2$. Dále předpokládejme, že $(S, stav)$ je konfigurace RVS odpovídající konfiguraci K_1 a $(S', stav')$ je konfigurace RVS odpovídající konfiguraci K_2 a navíc $(S, stav) \neq (S', stav')$. Potom $K_1 \vdash_M K_2$ právě tehdy když $(S, stav) \rightarrow (S', stav')$.*

Důkaz. ($\Rightarrow$) Mohou nastat následující 3 možnosti:

(a) $m = n$ a $\gamma' = s_b^n(\gamma)$ a $\delta_T(q, \gamma_n) = (q', b)$.
V tomto případě $p_n \in P$ je proces, který má řízení a tedy $stav(p_n) = (q, \gamma_n, v_n, 1)$. Nehchť p_m je libovolný proces, který je s p_n spojen komunikačním kanálem $i \in \{0, 1\}$ a tedy $stav(p_m) = (q'', a, v_m, 0)$, kde $v_m = v_n \oplus 1$ nebo $v_m = v_n \ominus 1$. Potom existuje $(q, 4, 0) \in \Sigma$ a

$$\begin{aligned} stav'(p_n) &= \delta((q, \gamma_n, v_n, 1), ((q, 4, 0), i)) = (q', s_b^n(\gamma), v_n, 1) \\ stav'(p_m) &= \delta((q'', a, v_m, 0), ((q, 4, 0), i)) = (q'', a, v_m, 0). \end{aligned}$$

Tedy $(S, stav) \to (S, stav')$ a $(S, stav')$ je konfigurace RVS, která odpovídá konfiguraci TS M $(q', \gamma'\Delta^\omega, m)$.

(b) $m = n + 1$ a $\gamma' = \gamma$ a $\delta_T(q, \gamma_n) = (q', R)$.
Opět $p_n \in P$ je proces, který má řízení a tedy $stav(p_n) = (q, \gamma_n, v_n, 1)$. Nejprve předpokládejme, že p_n je spojen oběma komunikačními kanály. Potom p_m je libovolný proces, který je s p_n spojen komunikačním kanálem $i \in \{0, 1\}$ a jeho identifikátor $v_m = v_n \oplus 1$ a tedy $stav(p_m) = (q'', a, v_n \oplus 1, 0)$. Potom existuje $(q', v_n \oplus 1, 0) \in \Sigma$ a

$$\begin{aligned} stav'(p_n) &= (q', \gamma_n, v_n, 0) = \delta((q, \gamma_n, v_n, 1), ((q', v_n \oplus 1, 0), i)) \\ stav'(p_m) &= (q', a, v_n \oplus 1, 1) = \delta((q'', a, v_n \oplus 1, 0), ((q', v_n \oplus 1, 0), i)). \end{aligned}$$

Tedy $(S, stav) \to (S, stav')$ a $(S, stav')$ je konfigurace RVS, která odpovídá konfiguraci TS M $(q', \gamma'\Delta^\omega, m)$. Nyní uvažujme případ, kdy p_n není spojen s procesem, jehož identifikátor je $v_n \oplus 1$ (musíme tedy přidat nový proces p_m).

$$\begin{aligned} stav'(p_n) &= (q', \gamma_n, v_n, 0) = \delta((q, \gamma_n, v_n, 1), ((q', v_n, 1), i)) \\ stav'(p_m) &= (q', \Delta, v_n \oplus 1, 1) = \delta((q_0^T, \Delta, 4, 0), ((q', v_n, 1), i)). \end{aligned}$$

Opět tedy Tedy $(S, stav) \to (S', stav')$ a $(S', stav')$ je konfigurace RVS, která odpovídá konfiguraci TS M $(q', \gamma'\Delta^\omega, m)$

(c) $m = n - 1$ a $\gamma' = \gamma$ a $\delta_T(q, \gamma_n) = (q', L)$, $n > 0$.
Opět $p_n \in P$ je proces, který má řízení a tedy $stav(p_n) = (q, \gamma_n, v_n, 1)$ a p_m je proces, který je s p_n spojen komunikačním kanálem $i \in \{0, 1\}$ a pro jeho identifikátor platí $v_m = v_n \ominus 1$. Tedy $stav(p_m) = (q'', a, v_n \ominus 1, 0)$. Potom existuje $(q', v_n \ominus 1, 0) \in \Sigma$:

$$\begin{aligned} stav'(p_n) &= (q', \gamma_n, v_n, 0) = \delta((q, \gamma_n, v_n, 1), ((q', v_n \ominus 1, 0), i)) \\ stav'(p_m) &= (q', a, v_n \ominus 1, 1) = \delta((q'', a, v_n \ominus 1, 0), ((q', v_n \ominus 1, 0), i)). \end{aligned}$$

Tedy $(S, stav) \to (S, stav')$ a $(S, stav')$ je konfigurace RVS, která odpovídá konfiguraci TS M $(q', \gamma'\Delta^\omega, m)$.

($\Leftarrow$) Vzhledem k tomu, že $(S, stav) \neq (S', stav')$, tak spolu nemohly komunikovat žádné dva procesy, z nichž ani jeden nemá řízení. Jediná možnost komunikace mezi procesy bez řízení je totiž 4. pravidlo přechodové funkce a toto pravidlo nemění stav. Komunikace tedy musela proběhnout mezi dvěma procesy z nichž právě jeden proces má řízení. Vzhledem k tomu, jak je přechodová funkce δ definována, to znamená, že ze stavu a aktuálního symbolu, který reprezentuje aktivní proces existuje přechod v TS M. Tedy $K_1 \vdash_M K_2$ a nový stav RVS odpovídá konfiguraci TS K_2. □

Důsledek 1.1. *Nechť M je TS, $(q, \gamma\Delta^\omega, n)$, kde $\gamma \in \Gamma^*$ je jeho konfigurace a $(S, stav)$ je konfigurace RVS odpovídající konfiguraci $(q, \gamma\Delta^\omega, n)$. Potom $(q, \gamma\Delta^\omega, n)$ $\vdash_M^* (q_F^T, \gamma'\Delta^\omega, m)$, kde $\gamma' \in \Gamma^*$ právě tehdy když z konfigurace $(S, stav)$ je dosažitelný koncový stav.*

Dále předpokládejme, že existuje jednoznačné kódování stavu RVS pomocí symbolů z nějaké abecedy Σ. Kód konfigurace RVS S potom označuji jako $\langle(S, stav)\rangle$. (Například $\Sigma = \{0, 1, \#_1, \#_2\}$ a v kódu S jsou jednotlivé složky odděleny symbolem $\#_1$, kódování automatu pomocí řetězce $\{0, 1\}^*$ je podobné jako kódování TS. RVS od funkce *stav* potom může být oddělen symbolem $\#_2$ a kódování funkce *stav* může být provedeno tak, že jednotlivé procesy spolu s akt. stavy jsou odděleny symbolem $\#_1$).

Nyní k samotnému důkazu, že problém dosažitelnosti koncového stavu (DOS) je nerozhodnutelný. Uvažujme funkci $\sigma : \{0, 1, \#\}^* \to \Sigma^*$, která provádí redukci z problému náležitosti MP.

$$\text{DOS} = \{\langle(S, stav)\rangle |\ \text{Ze stavu } (S, stav) \text{ dosáhne S konc. konfigurace}\}$$

Funkce σ pro zadaný vstupní řetězec x funguje následovně:

- Pokud x není správně zformovanou instancí MP, vrať kód RVS, která nikdy nedosáhne koncového stavu.
- Pokud x je ve tvaru $x_1\#x_2$, kde x_1 je kód TS M a x_2 je kód jeho vstupu w, vygeneruj kód RVS, která odpovídá počáteční konfiguraci TS M, tedy $(q_0^T, \Delta w\Delta^\omega, 0)$.

Podle důsledku 1.1. platí: $x \in \text{MP} \Leftrightarrow \sigma(x) \in \text{DOS}$. Tedy problém dosažitelnosti koncového stavu z dané konfigurace RVS je nerozhodnutelný.