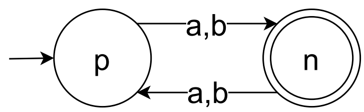
Teoretická informatika – Domáca úloha 1.

Michal Ľaš (xlasmi00)

3. novembra 2024

Príklad 1.

Minimálny DKA pre jazyk $L_1 = \{w \in \Sigma^* \mid |w| \mod 2 = 1\}$ vyzerá takto:



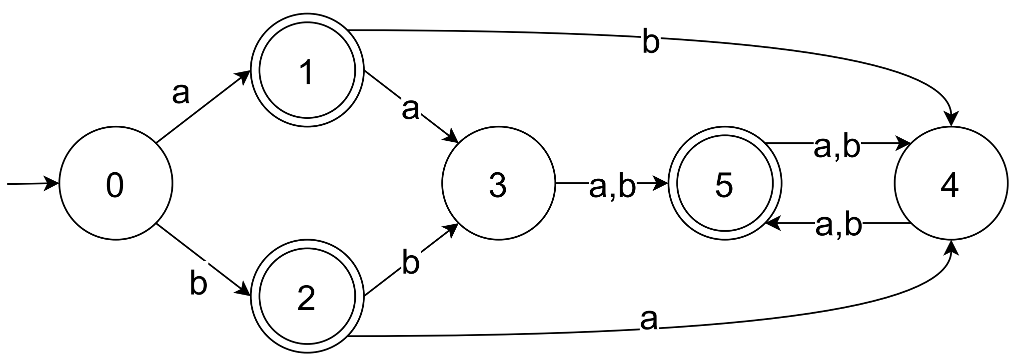
Tento automat je určite minimálny, pretože stav **p** je nekoncový a stav **n** je koncový, takže ich nie je možné zlúčiť do jedného stavu.

Jazyk prístupových reťazcov pre stavy **p** a **n**:

- $L_1^{-1}(p) = \{w \in \Sigma^* \mid |w| \mod 2 = 0\}$
- $L_1^{-1}(n) = \{w \in \Sigma^* \mid |w| \mod 2 = 1\}$

Teda rozklad $\Sigma^*_{/\sim_{L_1}} = \{L_1^{-1}(p), L_1^{-1}(n)\}$ a platí, že:
$\forall u, v \in \Sigma^* : u \sim_{L_1} v \Leftrightarrow$
$(u \in \{w \in \Sigma^* \mid |w| \mod 2 = 0\} \land v \in \{w \in \Sigma^* \mid |w| \mod 2 = 0\}) \lor$
$(u \in \{w \in \Sigma^* \mid |w| \mod 2 = 1\} \land v \in \{w \in \Sigma^* \mid |w| \mod 2 = 1\})$

Index $\sim_{L_1}$ je teda 2. Teraz je potrebné nájsť ekvivalentný DKA, ktorý nebude minimálny, počet jeho stavov bude súdeliteľný s 2 a práve jeden stav bude taký, že jeho jazyk prístupových reťazcov bude obsahovať práve dva prvky. Takýto automat by mohol vyzerať nasledovne:



Jazyky prístupových reťazcov pre jednotlivé stavy automatu:

- $L_1^{-1}(0) = \{\epsilon\}$
- $L_1^{-1}(1) = \{a\}$
- $L_1^{-1}(2) = \{b\}$
- $L_1^{-1}(3) = \{aa, bb\}$
- $L_1^{-1}(4) = \{w \in \Sigma^* \setminus \{\epsilon, a, b, aa, bb\} \mid |w| \mod 2 = 0\}$
- $L_1^{-1}(5) = \{w \in \Sigma^* \setminus \{\epsilon, a, b, aa, bb\} \mid |w| \mod 2 = 1\}$

Rozklad $\Sigma^*_{/\sim} = \{L_1^{-1}(0), L_1^{-1}(1), L_1^{-1}(2), L_1^{-1}(3), L_1^{-1}(4), L_1^{-1}(0)\}$ a platí, že:
$\forall u, v \in \Sigma^* : u \sim v \Leftrightarrow$
$(v \in \{\epsilon\} \wedge u \in \{\epsilon\}) \vee$
$(v \in \{a\} \wedge u \in \{a\}) \vee$
$(v \in \{b\} \wedge u \in \{b\}) \vee$
$(v \in \{aa, bb\} \wedge u \in \{aa, bb\}) \vee$
$(v \in \{w \in \Sigma^* \setminus \{\epsilon, a, b, aa, bb\} \mid |w| \mod 2 = 0\} \wedge u \in \{w \in \Sigma^* \setminus \{\epsilon, a, b, aa, bb\} \mid |w| \mod 2 = 0\}) \vee$
$(v \in \{w \in \Sigma^* \setminus \{\epsilon, a, b, aa, bb\} \mid |w| \mod 2 = 1\} \wedge u \in \{w \in \Sigma^* \setminus \{\epsilon, a, b, aa, bb\} \mid |w| \mod 2 = 1\})$

Splnenie podmienok

1. L_1 je zjednotením tried, ktoré sú vyznačené modro. Respektíve $L_1 = L_1^{-1}(1) \cup L_1^{-1}(2) \cup L_1^{-1}(5)$.
2. Index $\sim$ je **6**, čísla **2** a **6** sú súdeliteľné.
3. Trieda vyznačená červene ($L_1^{-1}(3)$) má práve 2 prvky.

Príklad 2. a)

Myšlienka: operácia $\Box L$ vytvára jazyk, ktorý obsahuje reťazce jazyka L také, že všetky prefixy tohto reťazca patria tiež do L. Pre každý regulárny jazyk existuje DKA, ktorý ho akceptuje. Skúsim teda vytvoriť algoritmus, ktorý prevedie ľubovoľný DKA akceptujúci jazyk L na KA, ktorý akceptuje jazyk $\Box L$. Ak takýto algoritmus existuje, potom operácia $\Box$ je uzavretá na triede regulárnych jazykov.
Majme DKA $M_1 = (Q, \Sigma, \delta, q_0, F)$, nech regulárny jazyk L je akceptovaný týmto automatom. Teraz vytvorím nový automat $M_2 = (Q', \Sigma', \delta', q_0', F')$, ktorý bude akceptovať jazyk $\Box L$.
Algoritmus pre prevod M_1 na M_2 by mohol vyzerať nasledovne:

- $Q' = Q$
- $\Sigma' = \Sigma$
- $\delta' = \pi \in \delta \mid \forall p, q \in F, \forall a \in \Sigma : q \in \delta(p, a)$
- $q_0' = q_0$
- $F' = F$

Inak povedané: KA M_2 vznikne tak, že z DKA M_1 budú odstránené všetky prechody, ktoré vedú z nekoncových stavov alebo do nekoncových stavov. Zostanú len prechody, ktoré idú z koncového stavu do koncového stavu. Tým je zaručené, že ak M_2 akceptuje nejaké slovo w, tak akceptuje aj všetky jeho prefixy.
V KA M_2 môžu vzniknúť nedostupné stavy. Nedostupné stavy sa však dajú odstrániť a automat previesť na DKA, pretože každý KA sa dá previesť na ekvivalentný DKA. Za poznámku stojí, že ak $q_0 \notin F$, tak potom $L(M_2) = \emptyset$. Keďže existuje algoritmus na prevod z DKA akceptujúceho regulárny jazyk L na DKA akceptujúci jazyk $\Box L$, tak operácia $\Box$ je uzavretá na množine regulárnych jazykov.

Príklad 2. b)

Majme ľubovoľný rekurzívne vyčísliteľný jazyk L, potom musí existovať TS M taký, že $L(M) = L$, aby bola operácia $\Box$ uzavretá na triede rekurzívne vyčísliteľných jazykov, tak je nutné popísať konštrukciu TS T takého, že $L(T) = \Box L$.

1. TS T má na svojom vstupe reťazec $w \in \Sigma^*$, kde Σ je abeceda TS M.
2. TS T si bude na svoju pomocnú pásku postupne generovať všetky prefixy reťazca w, ktorých je konečne veľa. Začína sa prázdnym reťazcom (ϵ) a postupne sa budú kopírovaním symbolov z hlavnej pásky na vedľajšiu generovať na vedľajšej páske všetky prefixy.
3. Po vygenerovaní každého prefixu TS T pustí simuláciu TS M. Následne sa generuje ďalší prefix a proces sa opakuje.
4. Ak TS M akceptuje každý prefix reťazca w, potom TS T akceptuje reťazec w.
5. Ak niektorý prefix nebol akceptovaný alebo TS T cyklí, tak TS T neakceptuje reťazec w.
6. Keďže existuje popis TS, ktorý akceptuje $\Box L$ pre ľubovoľný rekurzívne vyčísliteľný jazyk, tak operácia $\Box$ je uzavretá na triede rekurzívne vyčísliteľných jazykov.

Príklad 3.

1. Predpokladajme, že jazyk L_3 je bezkontextový, potom musí platiť pumping lemma pre bezkontextové jazyky: L je bezkontextový $\Rightarrow (\exists k \in \mathbb{N}^+ : \forall z \in \Sigma^* : z \in L \wedge |z| \geq k \Rightarrow$
$(\exists u, v, w, x, y \in \Sigma^* : uvwxy = z \wedge |vx| > 0 \wedge |vwx| \leq k \wedge \forall i \in \mathbb{N} \quad uv^iwx^iy \in L))$

2. Uvažujem o ľubovoľnom $k \in \mathbb{N}^+$.

3. Pre každé takéto k zvoľme slovo $z = a^k b^{k^2}$. Určite platí, že $z \in L_3$ a $|z| \geq k$.

4. Pre každé takéto slovo uvažujme o všetkých možných rozdeleniach na 5 častí u, v, w, x, y, kde $uvwxy = z$, $|vx| > 0$, $|vwx| \leq k$.

5. Tieto rozdelenia môžeme zaradiť do troch skupín:

   a) časť vx obsahuje iba symboly a alebo iba symboly b.

   b) časť vx obsahuje symboly a aj symboly b, a zároveň časť v alebo x obsahuje symboly a aj b.

   c) časť vx obsahuje symboly a aj symboly b, a zároveň časť v obsahuje len symboly a a časť x obsahuje len symboly b.

6. prípad **a)**: voľbou $i = 0$ budú narušené počty symbolov a alebo symbolov b, takže ak toto nové slovo z' rozdelíme na dve časti $z' = pq \mid p \in a^* \wedge q \in b^*$, tak $|p|^2 \neq |q|$ a teda takýto reťazec už nebude patriť do L_3.

7. prípad **b)**: voľbou $i = 2$ sa naruší forma slova tak, že už nebude platiť, že novo vzniknuté slovo z' sa bude dať rozdeliť na dve časti $z' = pq \mid p \in a^* \wedge q \in b^*$.

8. Prípad **c)**: voľbou $i = 2$ sa narušia počty symbolov a a b tak, že pri rozdelení novo vzniknutého slova z' na dve časti $z' = pq \mid p \in a^* \wedge q \in b^*$, bude $|p|^2 \neq |q|$ a teda takýto reťazec už nebude patriť do L_3. Rozdelenie pre tento prípad vyzerá nasledovne:

   - $u = a^{k-\alpha-\gamma}$
   - $v = a^{\alpha}$
   - $w = a^{\gamma} b^{\delta}$
   - $x = b^{\beta}$
   - $y = b^{k^2-\beta-\delta}$

   Musí platiť, že $\alpha + \gamma + \beta + \delta \leq k \wedge 0 < \alpha \wedge 0 < \beta \wedge 0 \leq \gamma \wedge 0 \leq \delta$. Pre $i = 2$ by nové slovo z' vyzeralo nasledovne: $z' = a^{k-\alpha-\gamma} a^{2\alpha} a^{\gamma} b^{\delta} b^{2\beta} b^{k^2-\beta-\delta}$. Po úprave: $z' = a^{k+\alpha} b^{k^2+\beta}$. Aby toto slovo patrilo do jazyka musí platiť, že $|a^{k+\alpha}|^2 = |b^{k^2+\beta}|$ teda, že $(k+\alpha)^2 = k^2 + \beta$. Po úprave tejto rovnice dostaneme $2k\alpha + \alpha^2 = \beta$. Táto rovnica však nemá riešenie, pretože $\alpha > 0 \wedge k > 0$ a $\beta < k$, čo vyplýva z uvedených podmienok. Záver je, že pre $i = 2$ nemožno nájsť také rozdelenie aby z' patrilo do jazyka.

9. Takto je ukázané, že pre každé uvažované rozdelenie je možné nájsť $i \in \mathbb{N}$ také, že $uv^iwx^iy \notin L_3$.

10. Pre jazyk L_3 tak neplatí pravá strana pumping lemmatu pre bezkontextové jazyky a preto L_3 nie je bezkontextový jazyk.

Príklad 4.

Myšlienka: hľadáme pravidlá typu $A \to \alpha\beta\gamma$, kde α, γ sa dajú prepísať na akékoľvek terminálne symboly a β garantuje, že vygenerujeme vetu s reťazcom *abc*. Budeme potrebovať nasledovné **pomocné množiny neterminálov**:

- $N_a = A \in N \mid \exists u \in \Sigma^* : A \Rightarrow_G^* ua$
- $N_b = A \in N \mid A \Rightarrow_G^* b$
- $N_c = A \in N \mid \exists u \in \Sigma^* : A \Rightarrow_G^* cu$
- $N_{ab} = A \in N \mid \exists u \in \Sigma^* : A \Rightarrow_G^* uab$
- $N_{bc} = A \in N \mid \exists u \in \Sigma^* : A \Rightarrow_G^* bcu$
- $N_{abc} = A \in N \mid \exists u, v \in \Sigma^* : A \Rightarrow_G^* uabcv$

Obecná štruktúra algoritmu:

1. $N_x^0 = \emptyset, \quad i = 0$
2. $\dots$
3. Pokiaľ $N_x^{i+1} \neq N_x^i$, tak:
   - $i = i + 1$
   - go to step 2
4. $N_x = N_x^i$

Algoritmus pre $N_a, x = a$:
2. $N_a^{i+1} = \{A \in N \mid \exists(A \to \alpha) \in P : \alpha \in (N_t \cup \Sigma)^*(N_a^i \cup \{a\})N_\epsilon^*\}$

Algoritmus pre $N_b, x = b$:
2. $N_b^{i+1} = \{A \in N \mid \exists(A \to \alpha) \in P : \alpha \in N_\epsilon^*(N_b^i \cup \{b\})N_\epsilon^*\}$

Algoritmus pre $N_c, x = c$:
2. $N_c^{i+1} = \{A \in N \mid \exists(A \to \alpha) \in P : \alpha \in N_\epsilon^*(N_c^i \cup \{c\})(N_t \cup \Sigma)^*\}$

Algoritmus pre $N_{ab}, x = ab$:
2. $N_{ab}^{i+1} = \{A \in N \mid \exists(A \to \alpha) \in P : \alpha \in (N_t \cup \Sigma)^*(N_{ab}^i \cup \{ab\} \cup \{a\}N_b \cup N_a\{b\} \cup N_aN_b)N_\epsilon^*\}$

Algoritmus pre $N_{bc}, x = bc$:
2. $N_{bc}^{i+1} = \{A \in N \mid \exists(A \to \alpha) \in P : \alpha \in N_\epsilon^*(N_{bc}^i \cup \{bc\} \cup \{b\}N_c \cup N_b\{c\} \cup N_bN_c)(N_t \cup \Sigma)^*\}$

Algoritmus pre $N_{abc}, x = abc$:
2. $N_{abc}^{i+1} = \{A \in N \mid \exists(A \to \alpha) \in P : \alpha \in (N_t \cup \Sigma)^*(N_{abc}^i \cup \{abc\} \cup \{a\}N_{bc} \cup \{a\}N_bN_c \cup \{ab\}N_c \cup N_a\{bc\} \cup N_{ab}\{c\} \cup N_aN_b\{c\} \cup N_a\{b\}N_c \cup N_{ab}N_c \cup N_aN_{bc} \cup N_aN_bN_c)(N_t \cup \Sigma)^*\}$

Ilustrácia algoritmu na gramatike zo zadania:

Najskôr je potrebné spočítať množiny $N_t, N_\epsilon, N_a, N_b, N_c$, pomocou nich ďalej spočítať množiny N_{ab} a N_{bc} a na koniec finálnu množinu N_{abc}.

$N_t^0 = \emptyset$	$N_\epsilon^0 = \emptyset$	$N_a^0 = \emptyset$	$N_b^0 = \emptyset$	$N_c^0 = \emptyset$
$N_t^1 = \{U\}$	$N_\epsilon^1 = \{U\}$	$N_a^1 = \{S, T, R\}$	$N_b^1 = \{U, W\}$	$N_c^1 = \{S\}$
$N_t^2 = \{U, R, T, S, W\}$	$N_\epsilon^2 = \{U\}$	$N_a^2 = \{S, T, R\}$	$N_b^2 = \{U, W, T\}$	$N_c^2 = \{S\}$
$N_t^3 = \{U, R, T, S, W\}$			$N_b^3 = \{U, W, T\}$	

$N_{ab}^0 = \emptyset$	$N_{bc}^0 = \emptyset$	$N_{abc}^0 = \emptyset$
$N_{ab}^1 = \{T\}$	$N_{bc}^1 = \{R\}$	$N_{abc}^1 = \{S\}$
$N_{ab}^2 = \{T\}$	$N_{bc}^2 = \{R, S\}$	$N_{abc}^2 = \{S\}$
	$N_{bc}^3 = \{R, S\}$	

Príklad 5.