

Nech L_1, L_2 sú jazyky.

- a) Zistite a dokážte, koľko slov môže mať jazyk $L_1^* \cap L_2^*$.
 b) Dokážte, že $(L^*)^* = L^*$.

a) Riešenie:

Počet slov: Vieme, že $\varepsilon \in L_1^* \cap L_2^*$, lebo $\varepsilon \in L_1^*$ a súčasne $\varepsilon \in L_2^*$
 (Lebo $L_1^* = \{\cup L_1^i \mid i \geq 0\}$ a $L_1^0 = \{\varepsilon\}$, pre L_2^* podobne)

Nech potom množina $L_1^* \cap L_2^*$ obsahuje okrem ε aj nejaké ine slovo, napr. u .

Z toho dostaneme, že $u \in L_1^*$ a súčasne $u \in L_2^*$.

Ak $u \in L_1^*$, z toho dostaneme: $\exists i : u \in L_1^i$, ale z toho dostaneme aj to, že pre každé $j \in \mathbb{N}$ (prir. číslo) platí: $u^j \in (L_1^i)^j$

Už stačí len ukázať, že $(L_1^i)^j \in L_1^*$

Mierne neformálne:

i -ráz

$(L_1^i)^j = L_1^i \cdot L_1^i \cdot \dots \cdot L_1^i$ (j -krát) $= (L_1 \cdot L_1 \dots L_1) \cdot (L_1 \cdot L_1 \dots L_1) \cdot \dots \cdot (L_1 \cdot L_1 \dots L_1) = L_1^{i \cdot j}$,

ale vieme, že $i \cdot j \in \mathbb{N}$, a preto $L_1^{i \cdot j} \in L_1^*$

Takže, keď $u \in L_1^*$, potom aj $u^j \in L_1^*$; keďže $u \in L_2^*$, potom aj $u^j \in L_2^*$.

ZÁVER: $u^j \in L_1^*$ a súčasne $u^j \in L_2^*$, preto $u^j \in L_1^* \cap L_2^*$, takže ak prienik obsahuje okrem ε aj nejaký iný prvok, tak ich obsahuje nekonečne veľa.

b) Riešenie:

$(L^*)^* = L^*$.

a) $(L^*)^* \subseteq L^*$

Nech $u \in (L^*)^*$, potom $\exists i : u \in (L^*)^i$, a teda $\exists i_1, \dots, i_j : u \in L^{i_1} \cdot L^{i_2} \cdot \dots \cdot L^{i_j}$, z čoho dostávame, že $u \in L^n$, kde $n = \sum i_i$ a keďže $L^n \subseteq L^*$, tak $u \in L^*$

b) $L^* \subseteq (L^*)^*$

$(L^*)^* = \{ (L^*)^0, (L^*)^1, (L^*)^2, \dots \}$, z toho vidíme platnosť inklúzie

Rozhodnite, či platí.

a) $L_1^* \cdot L_2^* = (L_1 \cdot L_2)^*$

b) $L_1^* \cdot L_2^* \cdot L_1^* = (L_1 \cdot L_2)^*$

c) $(L_1^* \cdot L_2^* \cdot L_1^*)^* = (L_1 \cdot L_2)^*$

Riešenie:

a) Neplatí, kontra-príklad :

$$L_1 = \{a\} \rightarrow L_1^* = \{\varepsilon, a, a^2, a^3, \dots\}$$

$$L_2 = \{b\} \rightarrow L_2^* = \{\varepsilon, b, b^2, b^3, \dots\}$$

$$L_1 \cdot L_2 = \{ab\} \rightarrow (L_1 \cdot L_2)^* = \{ab, abab, (ab)^3, \dots\}$$

$$L_1^* \cdot L_2^* = \{a^i b^j \mid i, j \geq 0\} = \{\varepsilon, a, b, ab, aab, abb, \dots\} \quad // !!$$

Ako kontraslovo sa dá použiť a,b apod.

b) Neplatí: Stačí použiť výsledok a), lebo $L_1^* \cdot L_2^* \subseteq L_1^* \cdot L_2^* \cdot L_1^*$ (stačí za L_1^* položiť a)
a ak vezmeme množiny L_1, L_2 z a), tak $a \in L_1^* \cdot L_2^* \cdot L_1^*$ ale súčasne $a \notin (L_1 \cdot L_2)^*$

Neplatí : Stačí použiť b), lebo $(L_1^* \cdot L_2^* \cdot L_1^*)^1 = L_1^* \cdot L_2^* \cdot L_1^* \subseteq (L_1^* \cdot L_2^* \cdot L_1^*)^*$,
Ak vezmeme množiny L_1, L_2 z a) tak $a \in (L_1^* \cdot L_2^* \cdot L_1^*)^*$ ale súčasne $a \notin (L_1 \cdot L_2)^*$

Rozhodnite, či platí.

- a) Ak $L = L^*$, tak $L = L^2$
- b) Ak $L = L^2$, tak $L = L^*$
- c) $L^* = (L^2)^* \cup (L \cdot (L^2)^*)$

Riešenie :

a) Platí

Podmienku $L=L^*$ si rozpíšeme na $L = \{\varepsilon, L, L^2, L^3, \dots\}$, tú môžeme rozpísať na 2 inklúzie:

$$L \subseteq \{\varepsilon, L, L^2, L^3, \dots\}, \text{ súčasne } \{\varepsilon, L, L^2, L^3, \dots\} \subseteq L \rightarrow L^2 \subseteq L$$

Z podmienky $L=L^*$ a $\varepsilon \in L^*$ vyplýva, že $\varepsilon \in L$.

Keďže $L^2=L.L$ a tiež $\varepsilon \in L$, potom $L.\{\varepsilon\} \in L^2 \Rightarrow L.\{\varepsilon\} \subseteq L^2 \rightarrow L \subseteq L^2$

Dokázali sme že pri podmienke $L=L^*$ platí $(L^2 \subseteq L \wedge L \subseteq L^2) \rightarrow L=L^2$

b) Neplatí. Ak $L=\emptyset$, tak $L=L^2=\emptyset$, ale $L^*=\{\varepsilon\} \neq \emptyset$ //!!!

c) Platí

Chceme : $L^* \subseteq (L^2)^* \cup (L \cdot (L^2)^*)$, $L^* = \{\varepsilon, L^1, L^2, L^3, \dots, L^{2^i}, L^{2^{i+1}}\}$,
potrebujeme ukázať, že každá z týchto množín je podmnožinou $(L^2)^* \cup (L \cdot (L^2)^*)$

Nech máme množinu tvaru L^{2^i} , $i \geq 0$ potom $L^{2^i} = (L^2)^i \in (L^2)^*$

Nech máme množinu tvaru $L^{2^{i+1}}$, $i \geq 0$ potom $L^{2^{i+1}} = L.(L^2)^i \in L.(L^2)^*$

Z toho dostávame, že $L^* \subseteq (L^2)^* \cup (L \cdot (L^2)^*)$

Chceme : $(L^2)^* \cup (L \cdot (L^2)^*) \subseteq L^*$

$$(L^2)^i = L^{2^i} \subseteq L^*$$

$$L.(L^2)^i = L^{2^{i+1}} \subseteq L^*$$

Dostali sme platnosť oboch inklúzií a teda platnosť rovnosti.

Majme homomorfizmus h nad abecedou $\Sigma = \{a\}$ taký, že $h(a) = \varepsilon$. Nech $L = \{a\}$.

Porovnajte množiny:

a) $h^{-1}(L^*)$ a $h(L^*)$

b) $h^{-1}(L^+)$ a $h(L^+)$

a) $L^* = \{ \varepsilon, a, aa, aaa, \dots \}$,

$h(L^*) = \{ \varepsilon \}$ // $h(\varepsilon) = \varepsilon$ (z pr.2.2.2 na st.8 vo Foriskovich skriptach-ver.13.4.04)

$h^{-1}(L^*) = \{ \varepsilon, a, aa, aaa, \dots \}$ (každý prvok sa zobrazí na ε , inverzný k a neexistuje)

$h(L^*) \subseteq h^{-1}(L^*)$

b) $h(L^+) = \{ \varepsilon \}$

$h^{-1}(L^+) = \emptyset$ (keďže $L^+ = \{a, aa, aaa, \dots\}$, na „a“ sa nič nezobrazí)

$h^{-1}(L^+) \subseteq h(L^+)$

- a) Rozhodnite, či platí: ak $L^2 = L^3$, tak $L = L^2$.
 b) Nech h je homomorfizmus, L_1, L_2 jazyky. Porovnajme množiny $h(L_1 \cap L_2)$ a $h(L_1) \cap h(L_2)$.

Riešenie:

- a) Neplatí, kontrapríklad : Nech $L = \{a\}^* - \{aa\}$

Potom $L^2 = \{a\}^*$, $L^3 = \{a\}^*$ a teda $L^2 = L^3$, ale evidentne $L^2 \neq L$ ($aa \in L^2$ ale $aa \notin L$)

- b) Môže ale nemusí nastať rovnosť, jedna inklúzia platí vždy:

i. Ak $L_1 = L_2$, tak $h(L_1 \cap L_2) = h(L_1) \cap h(L_2)$.

ii. Nech $L_1 = \{ab\}$, $L_2 = \{ac\}$ a $h(a) = a$
 $h(b) = \varepsilon$
 $h(c) = \varepsilon$

Potom $h(L_1) \cap h(L_2) \subsetneq h(L_1 \cap L_2)$

iii. Inklúzia $h(L_1 \cap L_2) \subseteq h(L_1) \cap h(L_2)$ musí nastať, lebo ak $u \in L_1 \cap L_2$, potom platí, že $u \in L_1$ a súčasne $u \in L_2$, preto $h(u) \in h(L_1)$ a súčasne $h(u) \in h(L_2)$, ale z toho vyplýva, že $h(u) \in h(L_1) \cap h(L_2)$, a to platí pre každý prvok $u \in L_1 \cap L_2$ a preto inklúzia platí.

Nech h je homomorfizmus, L_1, L_2 jazyky. Porovnajme množiny:

a) $h^{-1}(L_1 \cap L_2)$ a $h^{-1}(L_1) \cap h^{-1}(L_2)$

b) $h^{-1}(L_1 \cup L_2)$ a $h^{-1}(L_1) \cup h^{-1}(L_2)$

Riešenie :

a) $h^{-1}(L_1 \cap L_2) = h^{-1}(L_1) \cap h^{-1}(L_2)$

Platí : $h^{-1}(L_1 \cap L_2) \subseteq h^{-1}(L_1) \cap h^{-1}(L_2)$, lebo

Dôkaz:

$$\begin{aligned} \forall u \in h^{-1}(L_1 \cap L_2) : \exists v \in L_1 \cap L_2 : h(u) = v &\Rightarrow v \in L_1 \wedge v \in L_2 \Rightarrow h^{-1}(v) \in h^{-1}(L_1) \wedge h^{-1}(v) \in h^{-1}(L_2) \Rightarrow \\ \Rightarrow h^{-1}(v) \in h^{-1}(L_1) \cap h^{-1}(L_2) &\Rightarrow u \in h^{-1}(L_1) \cap h^{-1}(L_2) \end{aligned}$$

Slovne : Ak je nejaký prvok u z množiny $h^{-1}(L_1 \cap L_2)$, potom vieme, že obraz u leží v množine $L_1 \cap L_2$, tj. existuje prvok v z množiny $L_1 \cap L_2$ taký, že $h(u) = v$, z toho ale vyplýva, že prvok v leží v množine L_1 a súčasne v množine L_2 . Keďže v leží v množine L_1 , tak jeho vzor musí ležať v množine $h^{-1}(L_1)$, keďže v leží v množine L_2 tak jeho vzor leží v množine $h^{-1}(L_2)$. Z toho dostávame, že vzor v leží v množine $h^{-1}(L_1) \cap h^{-1}(L_2)$, a jeho vzorom je práve u .

Pozn. Aj keď uvádzam že vzorom je práve u , môžeme za u považovať celú triedu vzorov, pretože v prípade neinjektívneho homomorfizmu sa mi môže na jeden prvok zobrazit' veľa prvkov (tj. stačí keď sa nejaké písmeno zobrazí na ε). V dôkaze to nerobí problémy, lebo všetky sa zobrazia práve na jeden prvok a jeho vzorom je množina práve všetkých týchto prvkov.

Platí : $h^{-1}(L_1) \cap h^{-1}(L_2) \subseteq h^{-1}(L_1 \cap L_2)$

Dôkaz :

$$\begin{aligned} \forall u \in h^{-1}(L_1) \cap h^{-1}(L_2) : u \in h^{-1}(L_1) \wedge u \in h^{-1}(L_2) &\Rightarrow h(u) \in L_1 \wedge h(u) \in L_2 \Rightarrow h(u) \in L_1 \cap L_2 \Rightarrow \\ \Rightarrow u \in h^{-1}(L_1 \cap L_2) \end{aligned}$$

b) $h^{-1}(L_1 \cup L_2) = h^{-1}(L_1) \cup h^{-1}(L_2)$

Platí : $h^{-1}(L_1) \cup h^{-1}(L_2) \subseteq h^{-1}(L_1 \cup L_2)$

Dôkaz :

$$\begin{aligned} \forall u \in h^{-1}(L_1) \cup h^{-1}(L_2) \Rightarrow : u \in h^{-1}(L_1) \vee u \in h^{-1}(L_2) &\Rightarrow h(u) \in L_1 \vee h(u) \in L_2 \Rightarrow h(u) \in L_1 \cup L_2 \Rightarrow \\ \Rightarrow u \in h^{-1}(L_1 \cup L_2) \end{aligned}$$

Platí $h^{-1}(L_1 \cup L_2) \subseteq h^{-1}(L_1) \cup h^{-1}(L_2)$

Dôkaz

$$\begin{aligned} \forall u \in h^{-1}(L_1 \cup L_2) : \exists v \in L_1 \cup L_2 : h(u) = v &\Rightarrow v \in L_1 \vee v \in L_2 \Rightarrow h^{-1}(v) \in h^{-1}(L_1) \vee h^{-1}(v) \in h^{-1}(L_2) \Rightarrow \\ \Rightarrow h^{-1}(v) \in h^{-1}(L_1) \cup h^{-1}(L_2) &\Rightarrow u \in h^{-1}(L_1) \cup h^{-1}(L_2) \end{aligned}$$

Pozn. Dôkaz druhej časti je relatívne triviálny, stačí v dôkazoch prvej časti zameniť prieniky za zjednotenia a symbol \wedge za symbol \vee (pravda, až po zvážení, že to skutočne platí ☺)

Nech $L = \{a\}$ je jazyk nad abecedou $\Sigma = \{a\}$, nech h je homomorfizmus zo Σ^* do Σ^* taký, že $h(a^3) = \varepsilon$. Zistite, ako vyzerajú nasledujúce jazyky:

- a) $L_1 = (h^{-1}(h(L)))^*$
- b) $L_2 = L^2 \cdot (h(L))^+$
- c) $L_3 = L \cdot h^{-1}(L)$

Riešenie : Skôr než začneme uvažovať nad jednotlivými úlohami je dobré si ukázať, na čo sa zobrazí $h(a)$. Vieme, že h je homomorfizmus, a preto $h(a^3) = h(a).h(a^2) = h(a).h(a).h(a)$. Keby $h(a) = a$, tak $h(a^3) = a^3$, a to je v spore so zadáním, preto jediná možnosť je : $h(a) = \varepsilon$.

Potom jednotlivé úlohy majú toto riešenie:

$$L_1 = (h^{-1}(\{\varepsilon\}))^* = (\{\varepsilon, a, a^2, a^3, \dots\})^* = \{\varepsilon, a, a^2, a^3, \dots\}$$

$$L_2 = L^2 \cdot (h(L))^+ = \{aa\} \cdot (\{\varepsilon\})^+ = \{\varepsilon, aa\}$$

$$L_3 = \{a\} \cdot \emptyset = \emptyset (= \{\}, \text{ lebo je to množina prvkov } u.v, \text{ kde } u \in \{a\} \text{ a } v \in \{\}, \text{ ale také } v \text{ neexistuje})$$

Nájdite čo najjednoduchší množinový zápis jazyka generovaného gramatikou $G = (N, T, P, S)$, kde

$N = \{A, B, C, D, S\}$, $T = \{0, 1\}$,

$P = \{$

$S \rightarrow A \mid B$

$A \rightarrow C1C1CA \mid \varepsilon$

$B \rightarrow BD0D0D \mid \varepsilon$

$C \rightarrow 0C \mid \varepsilon$

$D \rightarrow D1 \mid \varepsilon$

$\}$. (Hint: čo najstručnejšie popíšte slová jazyka L , tento popis potom prepíšete do formálneho množinového zápisu.) Svoje tvrdenie zdôvodnite.

Riešenie : $L(G) = \{w \mid w \in \{a,b\}^*, \#_0(w)=2k \text{ alebo } \#_1(w)=2k, k>0\}$

Slovne : gramatika generuje slová z množiny $\{0,1\}^*$, ktoré obsahujú buď párny počet núl alebo párny počet jedničiek.

Zdôvodnenie :

Pravidlo $C \rightarrow 0C \mid \varepsilon$ znamená, že z neterminálu C vygenerujem ľubovoľný prvok množiny $\{0\}^*$, pravidlo $D \rightarrow D1 \mid \varepsilon$ znamená, že z neterminálu D vygenerujem ľubovoľný prvok množiny $\{1\}^*$.

Potom si treba všimnúť prvého pravidla $S \rightarrow A \mid B$, kde vidíme rozdelenie na dve vetvy A a B , pričom pri odvodzovaní z A používam iba A a C , a pri odvodzovaní z B používam iba B a D , takže každá vetva používa svoje dve pravidlá.

Keď sa pozrieme na pravidlá A : vidíme, že generujú výraz tvaru $\{C1C1C\}^*$.

Potom, ak vezmem slovo z tejto množiny $w \in \{C1C1C\}^*$, tak $w = u_1 u_2 \dots u_n$ pre nejaké n (nech $n > 0$) a pre nejaké $u_i \in \{C1C1C\}$. Z toho vidíme, že každé u_i obsahuje práve dve jedničky a medzi nimi ľubovoľne umiestnené 0, preto w obsahuje $2n$ jedničiek.

Pre odvodzovanie z B môžeme použiť podobne vytvorený postup.

Nájdite čo najjednoduchší množinový zápis jazyka generovaného gramatikou $G = (N, T, P, S)$, kde $N = \{A, B, C, S, P\}$, $T = \{a, b, c\}$,

$P = \{$

$S \rightarrow AS \mid P$

$P \rightarrow BP \mid CP$

$BA \rightarrow AB$

$AB \rightarrow BA$

$Ba \rightarrow ba$

$Aa \rightarrow aa$

$Ac \rightarrow ac$

$Cc \rightarrow cc$

$CP \rightarrow c$

$\}$. Svoje tvrdenie zdôvodnite.

Riešenie :

$$L\{G\} = \{a^i c^j \mid i \geq 0, j \geq 1\} \cup \{b^e a^i c^j \mid i, j \geq 1, e \in \{1, 0\}\}$$

Slovne : Gramatika vygeneruje ľubovoľnú postupnosť a a postupnosť aspoň jedného c, a ak sa v ňom nachádza a, tak sa na začiatku môže nachádzať b.

Postup : Najskôr si môžeme vytvoriť množinu všetkých slov z neterminálov. V prvom pravidle si všimneme možnosť vytvoriť množinu v tvare A^*S , ďalej sa dá pokračovať iba pomocou pravidla $S \rightarrow P$, takže dostaneme množinu A^*P . Pravidlá P so správajú podobne ako pre A , rozšíria mi možnosti na tvary $A^*\{B,C\}^*P$. Takže som schopný vytvoriť ľubovoľnú postupnosť A-čok, za ňou ľubovoľnú postupnosť B-ok a C-čok. Ešte si všimneme pravidiel $AB \rightarrow BA$ a $BA \rightarrow AB$, tie mi umožňujú ľubovoľne popresúvať A a B medzi sebou. Môžem preto tvrdiť, že sa dá vytvoriť ľubovoľné slovo v tvare $\{A,B\}^*\{B,C\}^*P$. Stačí si uvedomiť, že množina slov tvaru $\{B,C\}$ môže mať na začiatku požadovaný (akýkoľvek) počet B a tie som schopný umiestniť ľubovoľne medzi A.

Potrebujem teraz zistiť, pre akú podmnožinu slov v našom tvare som schopný nájsť odvodenie vedúce k terminálnemu slovu.

Ako je vidno, terminály dostanem za použitia pravidiel, ktoré na ľavej strane majú aj terminál, preto sú všetky okrem jedného na začiatku nepoužiteľné. Ako prvé musím použiť pravidlo $CP \rightarrow c$, keď mi vznikne neterminál a umožní mi použiť ďalšie dve pravidlá. Znamená to ale aj, že každé slovo, ktoré sa má dať previesť na neterminály, musí obsahovať CP a keďže P sa vyskytuje vždy iba na konci, tak regulárne slová sú v tvare wCP . Slovo tvaru wCP môžeme previesť na wc . Aby sme toto mohli ukončiť, muselo by w byť ε alebo w musí na konci obsahovať A alebo C, (S, P nebudem uvažovať, lebo CP dostanem až po ich odstránení). Ak by sa tam nachádzalo B, dalo by sa pokračovať, ak by pred ním bolo A a urobili by sme ich vzájomnú výmenu, ale ten prípad je zahrnutý v tvare $\{A,B\}^*\{B,C\}^*P$, keď za $(B,C)^*$ vyberieme ε .

Rozoberme nasledujúce prípady:

- ak máme slovo v tvare $uCCP$, tak sme schopní ho previesť na tvar ucc , a tak isto môžeme pokračovať, ak tam je viac C.
- ak máme slovo v tvare $uACP$, tak ho môžeme previesť na uac , ale tu prichádzame k obmedzeniu ďalšej voľby, a to na koniec u. Tam sa môže nachádzať A a tak by sme mohli pokračovať, alebo tam môže byť B. Ak by to bolo B, máme slovo v tvare $vBACP$, ktoré sa odvodí na $vbac$. Nech ale končí v na ľubovoľný neterminál, ten sa už nebude dať previesť na terminál.

Zhrnutím týchto možností dostávame množinu uvedenú ako riešenie.

Zostrojte bezkontextovú gramatiku, ktorá bude generovať jazyk všetkých dobre uzátvorkovaných výrazov nad abecedou $\Sigma = \{ (,), [,] \}$.

(T.j. v $L(G)$ majú byť napr. slová ϵ , $(())$, $([])$, ale nie $([)$, $(($ ani $][.)$)
Dokážte správnosť vašej konštrukcie

Riešenie : $G = (\{S\}, \{ (,), [,] \}, P, S)$
 $P = \{ S \rightarrow \epsilon \mid SS \mid (S) \mid [S] \}$

To, že gramatika generuje správne riešenie je zrejmé, pravidlo $S \rightarrow SS$ použijeme, keď chceme dostať 2 zátvorky za sebou, inak použijeme zvyšné pravidlá na vnorenie zátvoriek do seba alebo pre ukončenie generovania v danej časti, tak isto je vidno, že ľubovoľný výraz sa dá vygenerovať tak, že vezmeme výraz a ten budem mať určitý počet „prvoúrovňových“ zátvoriek, tj. bude tvaru $(\text{výraz})(\text{výraz})\dots(\text{výraz})$. Tak si nagerujeme dostatočnú postupnosť S za sebou a postupne vytvárame vnútra zátvoriek.

Ešte to ale treba dokázať, tj. dokázať, že generuje iba zadaný jazyk a že dokáže vygenerovať ľubovoľný uzátvorkovaný výraz:

a) že generuje iba správne uzátvorkované výrazy :

Indukciou vzhľadom na dĺžku odvodu.

IP „: Každá vetná forma odvodená na m krokov je buď správne uzátvorkovaný výraz alebo obsahuje symboly S , ktoré sa nachádzajú medzi zátvorkami (ktoré spolu „sedia“ alebo mimo zátvoriek ,tj. je tvaru uwS^nv , alebo wS^n alebo S^nw , kde w je dobrý výraz a u, v tvoria v tomto poradí správne uzátvorkovanie (dá sa tvrdiť, že uv je správne uzátvorkovaný výraz). Výraz ϵ je korektný.

Pre $m=0$: každé odvodzovacie pravidlo prevedie S na korektný výraz

Predpokladajme platnosť IP pre $m=k$, chceme platnosť pre $m = k + 1$.

Podľa IP je slovo v niektorom z tvarov:

- 1) už je správne uzátvorkovaný výraz, takže je to v poriadku, ďalšie odvodzovanie nie je možné
- 2) $uwS^nv = uwSS^{n-1}v \rightarrow uwS^{n-1}v$
 $\rightarrow uSwSS^{n-1}v = uS^{n+1}v$
 $\rightarrow uw(S)S^{n-1}v$ - tvar uwS^nv
 $\rightarrow u[S]S^{n-1}v$ - podobný prípad
- 3) tvar wS^n
 $\rightarrow wS^{n-1}$
 $\rightarrow wS^{n+1}$
 $\rightarrow w(S)S^{n-1}$ - tvar wS^n
 $\rightarrow w[S]S^{n-1}$ - podobne
- 4) tvar S^nw bude vyzerat' obdobne

Z každého tvaru môžeme odvodiť iba dobrý tvar, takže máme dokázané, že vieme odvodiť iba správne tvary a teda naša gramatika generuje iba jazyk L .

b) Potrebujeme dokázať že G vygeneruje každé slovo z L , dokážeme indukciou na dĺžku odvodeného slova.

IP : Vieme odvodiť správne uzátvorkovaný výraz dĺžky $2k$ (zátvorky sa párujú)

Pre $n = 0$, tj. $|w|=0$, $w = \epsilon$, ϵ vieme odvodiť, takže tvrdenie platí

Pre $n = 2(k+1)$, $|u| = 2k+1$, tak môžu nastať prípady :

u je tvaru (v) alebo $[v]$, tak použijeme pravidlo $S \rightarrow (S)$ alebo $S \rightarrow [S]$, a keďže $|v| = k$, vieme ho podľa IP odvodiť, preto po dosadení dostaneme (v) alebo $[v]$

u je tvaru vw ($v \neq \epsilon$, $w \neq \epsilon$, tj. $|v| \geq 2, |w| \geq 2$) a $|v| \leq k, |w| \leq k$, keď použijeme pravidlo $S \rightarrow SS$ a z IP vieme odvodiť v aj w tak vieme odvodiť z $SS \rightarrow^* vS \rightarrow^* vw = u$.

Záver : Ekvivalencia je dokázaná.

Zostrojte regulárnu gramatiku, ktorá bude generovať jazyk všetkých telefónnych čísel do mobilných sietí.

(Jazyk je nad abecedou $\Sigma = \{0, \dots, 9, /\}$, príkladom slova z neho je 0907/123456.)

Dokážte správnosť vašej konštrukcie. Koľko najmenej neterminálov potrebujete?

Riešenie : vytvoríme gramatiku $G = (\{S\}, \{0, \dots, 9, /\}, P, S)$, kde

$P = \{ S \rightarrow 0900/000000 \mid 0900/0001 \dots 0909/999999 \}$

Dokazovanie správnosti nie je nutné, je vidno z konštrukcie. Použili sme iba jeden neterminál.

Pozn. Využili sme faktu, že množina všetkých telefónnych čísel do mobilných sietí je konečná (jazyk L obsahuje konečný počet slov) a preto tento jazyk môžeme generovať priamo a množina pravidiel bude konečná, čo je podmienka pre frázové gramatiky.

Daná je gramatika $G = (N, T, P, S)$, kde $N = \{S, A, B, C, D, F, P\}$, $T = \{a, b\}$ a

$$P = \{ \begin{array}{l} S \rightarrow aC \mid ASA \mid bb \\ A \rightarrow AC \mid AAC \mid SFC \\ B \rightarrow Da \\ C \rightarrow ab \mid CaC \\ D \rightarrow BB \mid Sa \\ F \rightarrow FC \mid A \\ P \rightarrow aAbP \mid AA \mid P \end{array} \}$$

Konštrukciou z prednášky odstráňte z tejto gramatiky všetky nedosiahnuteľné neterminály.

Riešenie :

Dosiahnuteľné neterminály : $H_0 = \{S\}$
 $H_1 = \{C, A\} \cup H_0$
 $H_2 = \{F\} \cup H_1 = H = \{S, A, C, F\}$

Postup : V prvom kroku môžeme dosiahnuť len S, v ďalšom z S môžeme dosiahnuť aj C a A a v ďalšom kroku z A môžeme dosiahnuť F, tým postup končí.

Nová gramatika je : $G' = (N', T', P', S')$, kde $N' = \{S, A, C\}$, $T' = T$, $S' = S$ a

$$P' = \{ \begin{array}{l} S \rightarrow aC \mid ASA \mid bb \\ A \rightarrow AC \mid AAC \mid SFC \\ C \rightarrow ab \mid CaC \\ F \rightarrow FC \mid A \end{array} \}$$

Gramatika G' generuje ten istý jazyk ako G , tj. $L(G') = L(G)$. Je to dané tým, že sme odstránili iba nedosiahnuteľné neterminály a ich pravidlá, takže nemohol nastať prípad, že by tieto pravidlá boli použité pri nejakom generovaní určitého slova.

Pozn. Je vhodné si všimnúť, že stačilo vždy odstrániť celý riadok pravidiel, a nenastal prípad nutnosti odstraňovania iba určitej časti pravidiel z jedného riadku.

Daná je gramatika $G = (N, T, P, S)$, kde $N = \{S, A, B, C, D, F, P\}$, $T = \{a, b\}$ a

$P = \{$
 $S \rightarrow aC \mid ASA \mid bb$
 $A \rightarrow AC \mid AAC \mid SFC$
 $B \rightarrow Da$
 $C \rightarrow ab \mid CaC$
 $D \rightarrow BB \mid Sa$
 $F \rightarrow FC \mid A$
 $P \rightarrow aAbP \mid AA \mid P$
 $\}$

Konstruáciou z prednášky odstráňte z tejto gramatiky všetky neterminály, z ktorých sa nedá odvodiť terminálne slovo

Riešenie:

Zostrojíme množinu neterminálov, z ktorých sa dá zostrojiť terminálne slovo.

$$H_0 = \{S, C\}$$

$$H_1 = \{D\} \cup H_0$$

$$H_2 = \{B\} \cup H_1 = H = \{S, B, C, D\}$$

Postup : Postupne pridávame neterminály, z ktorých sa dá určite urobiť terminálne slovo. V prvom kroku to sú iba S a C vďaka pravidlám $S \rightarrow bb$ a $C \rightarrow ab$, tj. Pravidlá, kde sa vpravo objavujú iba terminály, lebo o žiadnom neterminále nevieme povedať, či sa dá previesť na terminál. V ďalšom kroku už môžeme pridať D vďaka pravidlu $D \rightarrow Sa$, lebo a je terminál a o S už vieme, že sa dá na terminál previesť. Takto pokračujeme, až kým neprestranú pribúdať neterminály.

Pozn. Je vhodné si všimnúť neúčast' A. To potrebuje k ukončeniu v dvoch pravidlách seba, v treťom F, ale F potrebuje v druhom A, takže by vznikol nekonečný cyklus dvoch neterminálov a v prvom seba, tak isto by vznikol neukončiteľný cyklus.

Nová gramatika je : $G' = (N', T', P', S')$, kde $N' = \{S, B, C, D\}$, $T' = T$, $S' = S$ a

$P' = \{$
 $S \rightarrow aC \mid bb$
 $B \rightarrow Da$
 $C \rightarrow ab \mid CaC$
 $D \rightarrow BB \mid Sa$
 $\}$

Pozn. Na rozdiel od úlohy 4.5, kde sa odstraňovali nedosiahnuteľné neterminály a kde sa mazali celé riadky pravidiel, tu je nutné skontrolovať korektnosť po odstránení neterminálov a ich pravidiel a odstrániť aj pravidlá, ktoré umožňujú ich vznik, v našom prípade to bolo pravidlo $S \rightarrow ASA$.

Pomocou konečného počtu použití homomorfizmu jazykov, inverzného homomorfizmu jazykov, zret'azenia, zjednotenia a prieniku zostrojte z jazyka

$$L_1 = \{w \mid w \text{ patrí do } \{a, b\}^* ; \#_a(w) = \#_b(w)\}$$

jazyk

$$L_2 = \{w \mid w \text{ patrí do } \{a, b, c\}^* ; \#_a(w) = \#_b(w) = \#_c(w)\}$$

$$L_1 \triangleleft \{a^n b^n \mid n \in \mathbb{N}_0\}$$

RIEŠENIE :

Definujeme homomorfizmy:

$$h_1(a) = \varepsilon$$

$$h_2(a) = a$$

$$h_1(b) = a$$

$$h_2(b) = \varepsilon$$

$$h_1(c) = b$$

$$h_2(c) = b$$

$$\text{Potom } L_2 = h_1^{-1}(L_1) \cap h_2^{-1}(L_1)$$

Prvý inverzný homomorfizmus mi vytvorí množinu $\{w \in \{a, b, c\}^* \mid \#_b(w) = \#_c(w)\}$, tj. Všetky slová s rovnakým počtom b a c a ľubovoľne povkladanými terminálmi (a aj ich ľubovoľným počtom, lebo $h_1(a) = \varepsilon$). Druhý vytvorí množinu $\{w \in \{a, b, c\}^* \mid \#_a(w) = \#_c(w)\}$. Ich prienik bude požadovaná množina, prepojenie je viazané na počet terminálov c, od ich počtu závisí počet a aj počet b.

Nech G je bezkontextová gramatika. Označme $D(G)$ gramatiku, ktorá vznikne z G odstránením nedosiahnuteľných neterminálov. Podobne označme $T(G)$ gramatiku, ktorá vznikne z G odstránením neterminálov, z ktorých nie je možné dosiahnuť terminálne slovo.

Porovnajte množiny $L(D(T(G)))$ a $L(G)$.

Porovnajte množiny $L(T(D(G)))$ a $L(G)$.

RIŠENIE :

Pomôže nám uvedomiť si, že po odstránení nedosiahnuteľných neterminálov nám vznikne “ekvivalentná” gramatika, lebo žiaden z terminálov sme nemohli použiť na generovanie nejakého slova, čo vyplýva z nedosiahnuteľnosti. Nemohli sme z nich vytvoriť ale ani neterminálne slovo, takže sa vôbec nepodielajú na generovaní slov.

Obtiažnejšie je to s odstránením neterminálov, z ktorých nie je možné dosiahnuť terminálne slovo. Jazyk, ktorý generuje nová gramatika je rovnaký, ale ubudli možnosti generovania neterminálnych slov.

Preto ak najskôr odstránime nedosiahnuteľné a potom neterminovateľné neterminály, generovaný jazyk sa nezmení.

Ak ale najskôr odstránime neterminovateľné a potom nedosiahnuteľné, ubudnú síce v prvom kroku nejaké slová zložené z neterminálov, ale tieto sa nedajú previesť na terminály a preto ich odstránením sa generovaný jazyk nezmení. Po odstránení nedosiahnuteľných neterminálov sa jazyk tiež nemení. Tento druhý postup je postupom pre vytvorenie gramatiky v normálnom redukovanom tvare.

Záver : $L(D(T(G))) = L(G)$.

$L(T(D(G))) = L(G)$.

Predpokladajte, že v špecifikácii programovacieho jazyka Pascal nie je povedané, že `else` sa vzťahuje k najbližšiemu `if`-u. Napíšte program, ktorý sa na rovnakom vstupe môže správať na rôznych kompilátoroch rôzne.

Riešenie:

Ide o príklad z prednášky, tj. o problém typu:

```
if a>5 then
  if b>5
    then
      writeln('ano');
  else
    writeln('nie');
```

No a to `else` by sa mohlo vzťahovať na prípady $a \leq 5$ ale aj $b \leq 5$, preto tá nejednoznačnosť. Rada na záver : nájdite si lepší programovací jazyk, ako je napr. C a máte po problémoch s riešením problému.

Je daná bezkontextová gramatika $G = (N, T, P, S)$, kde $N = \{S\}$, $T = \{a, b\}$,
 $P = \{ S \rightarrow aS \mid Sb \mid bSa \mid \epsilon \}$.

Dokážte, že gramatika je jednoznačná.

Riešenie : gramatika nie je jednoznačná , je nejednoznačná, jebo existujú dve krajné odvodenia jedného slova, a to napríklad slova ab :

$S \rightarrow aS \rightarrow aSb \rightarrow ab$

$S \rightarrow Sa \rightarrow aSb \rightarrow ab$,

kde prvý krok odvodenia je evidentne rozdielny, ale odvodené slovo je to isté, a vždy ide o krajné odvodenie (v našom prípade ľavé aj pravé súčasne)

Zostrojte frázovú gramatiku, ktorá bude generovať jazyk $L = \{a^{2^n} \mid n \geq 0\}$. Ukážte, ako sa dá vo vašej gramatike odvodiť ľubovoľné slovo z L . Stručne zdôvodnite, prečo sa žiadne iné slovo odvodiť nedá.

Riešenie :

Nech $G = (N, T, P, S)$, kde

$N = \{S, B, E, R\}$ - B jako BEGIN, R jako RUNER, E jako END

$T = \{a\}$

$P = \{ S \rightarrow BaE \mid \varepsilon$ - začneme, alebo ε
 $B \rightarrow BR \mid \varepsilon$ - začiatok vypúšťa bežca, alebo sa ukončí činnosť
 $Ra \rightarrow aaR$ - bežec duplikuje áčka pred sebou
 $RE \rightarrow E$ - na konci zlikvidujeme bežca
 $R \rightarrow \varepsilon$ - ukončujeme činnosť
 $\}$

Ak dostaneme slovo v tvare a^{2^n} $n > 0$, tak použijeme najskôr prvé pravidlo $S \rightarrow BaE$, potom n -krát vygenerujeme bežca, necháme ho prebehnúť slovom (3. pravidlo) a zdvojnásobiť vždy počty áčok a na konci ho vždy zrušíme, na konci tohoto cyklu zrušíme začiatok a koniec (B a E).

Dôvod, prečo sa nedá nič iné vygenerovať :

Buď na začiatku vygenerujeme ε , alebo použijeme pravidlo $S \rightarrow BaE$, ale potom keď vypustíme bežca, musí prejsť na koniec, lebo inde sa nedá zrušiť, a teda vždy zdvojnásobuje počet áčok.

Zostrojte frázovú gramatiku, ktorá bude generovať jazyk

$L = \{x\#y \mid x, y \text{ sú čísla v dvojkovej sústave také, že } x < y\}$. (L je jazyk nad abecedou $\{0, 1, \#\}$.) Stručne zdôvodnite, že vaša gramatika naozaj generuje tento jazyk - teda že viete odvodiť všetky slová z L a neviete odvodiť nič iné.

Riešenie : $G = (S, \{0, 1, \#\}, P, S)$

$T = \{S, B, R, A, T, E\}$ - B - BEGIN, R - RUNNER, E - END, A - ADVANCER

$P = \{ \quad S \rightarrow B0\#1E \quad \text{-vygenerujem minimálnu možnosť}$

$B \rightarrow BR \mid \varepsilon \quad \text{- vybehne bežec, alebo ukončenie generovania zmien}$

$R0 \rightarrow 0R \quad \text{- prechod bežca cez cifry}$

$R1 \rightarrow 1R$

$RE \rightarrow AE \mid E \quad \text{- bežec môže zvyšovať o 1 druhé slovo alebo ho nemeniť (keď je na konci)}$

$R\# \rightarrow A\#T \mid \#R \quad \text{- keď je bežec v strede, môže zvyšovať o 1 vpravo aj vľavo alebo prejsť}$

$0A \rightarrow 1 \quad \text{- zvýšenie 0 o 1 znamená zmenu na 1}$

$1A \rightarrow A0 \quad \text{- zvýšenie 1 o 1 znamená zmenu na 0 a zvýšenie o 1 prednú časť}$

$BA \rightarrow B1 \quad \text{- ak som skončil až na začiatku, treba dať na začiatok 1}$

$\#A \rightarrow \#1 \quad \text{-podobne v druhej časti, len začiatok je \# (11A -> 1A0 -> A10-> 111)}$

$T0 \rightarrow 0T \quad \text{-keďže treba zvyšovať od konca, v druhej časti prejde T na koniec a zmení sa na A}$

$T1 \rightarrow 1T$

$TE \rightarrow AB \quad \text{- zmena}$

$E \rightarrow \varepsilon\}$

Princíp je založený na zvyšovaní o 1 ľavej aj pravej časti súčasne, alebo len pravej. Na začiatku sa vygeneruje bežec, ktorý prejde do stredu, tam má 2 možnosti, buď prejsť na druhú stranu alebo dať signál na zvyšovanie o 1 pravej aj ľavej časti, tj vygeneruje symbol A na konci ľavej, a pomocný symbol T na pravej. T prejde na koniec pravej časti a zmení sa na A , kde zvýši pravú časť o 1. R málo možnosť prejsť cez stred a dojsť na koniec, tam sa zmeniť na A alebo zaniknúť.

To, že jazyk generuje práve jazyk L je dané jeho funkciou, lebo určite ľavá strana je aspoň o 1 menšia než pravá a keď ju zväčšíme, zväčší sa aj pravá. To, že generuje všetko je dané zvyšovaním ľavej strany o 1, takže môžeme ľavú stranu dostať na ľubovoľnú hodnotu a pravú následne na ľubovoľnú väčšiu.

ULOHA

Daná je bezkontextová gramatika $G = (N, T, P, S)$, pričom $N = \{S, A, B, C, D, F\}$, $T = \{a, b\}$,

$P = \{ S \rightarrow AaF \mid CbSbDaBaa \mid baADba$

$A \rightarrow \varepsilon \mid aB$

$B \rightarrow \varepsilon \mid ba \mid CaDaBbFba$

$C \rightarrow A \mid bcc \mid ABC$

$D \rightarrow ACB \mid CFB$

$F \rightarrow aFa \mid abaA \}$

Zostrojte takú bezkontextovú gramatiku G' , ktorá neobsahuje ε na pravej strane žiadneho pravidla (bezepsilonovu) a platí $L(G) = L(G')$. Použite konštrukciu z prednášky.

RIEŠENIE :

Keďže daná gramatika je bezkontextová a negeneruje ε , tak úloha je riešiteľná (nová gramatika nemôže vygenerovať ε)

Zostrojíme množinu miznúcich neterminálov, tj. tie, ktoré sa dajú previesť na ε alebo na postupnosť už miznúcich neterminálov :

$$H_0 = \{A, B\}$$

$$H_1 = \{C\} \cup H_0$$

$$H_2 = \{D\} \cup \text{lebo } D \rightarrow ACB \text{ \& } H_0 = \{A, B\} \cup H_1 = H = \{A, B, C, D\}$$

Zostrojím $G = (N', T', P', S') : N' = N, T' = T, S' = S,$

$$P' = \{ \xi \rightarrow u_1 \eta_1^{k_1} u_2 \eta_2^{k_2} \dots u_n \eta_n^{k_n} u_{n+1} \mid \xi \rightarrow u_1 \eta_1^{k_1} u_2 \eta_2^{k_2} \dots u_n \eta_n^{k_n} u_{n+1} \in P, \forall i : \eta_i \in H, u_i \in (N - H \cup T)^* \},$$

Toto je zostrojenie definované na prednáške.

P_1' môže vyzeráť takto (uvediem postup ako odstraňujem jednotlivé problematické pravidlá) :

Najskôr odstránim pravidlo $A \rightarrow \varepsilon$

$P_1' = \{ S \rightarrow AaF \mid CbSbDaBaa \mid baADba \mid aF \mid baDba$

-pribudli posledné 2, keď v prvom. a tretom.

A nahradím ε -om

$A \rightarrow aB$ // odstranili sme $A \rightarrow \varepsilon$

$B \rightarrow \varepsilon \mid ba \mid CaDaBbFba$

$C \rightarrow A \mid bcc \mid ABC \mid \varepsilon \mid BC$

- pridal som si pomocné ε (mohlo vzniknúť z

A), $C \rightarrow BC$ sme pridali, lebo mohlo vzniknúť takto v

povodnej gramat: $C \rightarrow ABC \xrightarrow{A \rightarrow \varepsilon} BC$

- atď.

$D \rightarrow ACB \mid CFB \mid CB$

$F \rightarrow aFa \mid abaA \mid aba\}$

P_2' môže vyzeráť takto (**odstraňujem $B \rightarrow \varepsilon$** z P_1') :

$P_2' = \{ S \rightarrow AaF \mid CbSbDaBaa \mid CbSbDaaa \mid baADba \mid aF \mid baDba$

$A \rightarrow aB \mid a$

$B \rightarrow ba \mid CaDaBbFba \mid CaDabFba$

$C \rightarrow A \mid bcc \mid ABC \mid \varepsilon \mid BC \mid AC \mid C$

$D \rightarrow ACB \mid CFB \mid CB \mid AC \mid CF \mid C$

$F \rightarrow aFa \mid abaA \mid aba\}$

P_3' môže vyzerat' takto (**odstraňujem $C \rightarrow \varepsilon$** ($C \in (H_1 - H_0)$)), vzniklo v prvom kroku kvôli postupnosti $C \rightarrow A \rightarrow \varepsilon$): :

$P_3' = \{$
 $S \rightarrow AaF \mid CbSbDaBaa \mid \text{bSbDaBaa} \mid baADba \mid aF \mid baDba \mid CbSbDaaa \mid bSbDaaa$
 $A \rightarrow aB \mid a$
 $B \rightarrow ba \mid CaDaBbFba \mid CaDabFba \mid aDaBbFba \mid aDabFba$
 $C \rightarrow A \mid bcc \mid ABC \mid BC \mid AC \mid C \mid AB \mid B \mid A$
 $D \rightarrow ACB \mid CFB \mid CB \mid AC \mid CF \mid C \mid AB \mid FB \mid B \mid A \mid \varepsilon$
 $F \rightarrow aFa \mid abaA \mid aba\}$

P_4' môže vyzerat' takto (**odstraňujem $D \rightarrow \varepsilon$** , ($D \in (H_2 - H_1)$)), vzniklo v predchádzajúcom kroku kvôli postupnosti $D \rightarrow C \rightarrow \varepsilon$): :

$P_4' = \{$
 $S \rightarrow AaF \mid CbSbDaBaa \mid baADba \mid aF \mid baDba \mid CbSbDaaa \mid bSbDaBaa \mid bSbDaaa \mid$
 $CbSbaBaa \mid baAba \mid baba \mid CbSbaaa \mid bSbaBaa \mid bSbaaa$
 $A \rightarrow aB \mid a$
 $B \rightarrow ba \mid CaDaBbFba \mid CaDabFba \mid aDaBbFba \mid aDabFba \mid CaaBbFba \mid CaabFba \mid aaBbFba \mid$
 $aabFba$
 $C \rightarrow A \mid bcc \mid ABC \mid BC \mid AC \mid C \mid AB \mid B \mid A$
 $D \rightarrow ACB \mid CFB \mid CB \mid AC \mid CF \mid C \mid AB \mid FB \mid B \mid A$
 $F \rightarrow aFa \mid abaA \mid aba\}$

P_4' už je hľadaná množina pravidiel, keďže ide o množinu, ešte treba odstrániť duplicitné zápisy v pravidlách (v predchádzajúcich krokoch boli tiež ponechané, v korektnom zápise aj pre zjednodušenie je vhodné ich vyhadzovať ihneď, pre overenie neduplicity sa dajú usporiadať abecedne, aspoň sčasti)

$P' = \{$
 $S \rightarrow AaF \mid aF \mid baADba \mid baDba \mid baAba \mid baba \mid bSbDaBaa \mid bSbDaaa \mid bSbaBaa \mid bSbaaa \mid$
 $CbSbDaBaa \mid CbSbDaaa \mid CbSbaBaa \mid CbSbaaa$
 $A \rightarrow aB \mid a$
 $B \rightarrow aDaBbFba \mid aDabFba \mid aaBbFba \mid aabFba \mid ba \mid CaDaBbFba \mid CaDabFba \mid CaaBbFba \mid$
 $CaabFba$
 $C \rightarrow A \mid AB \mid ABC \mid AC \mid bcc \mid B \mid BC \mid C$
 $D \rightarrow A \mid AB \mid AC \mid ACB \mid B \mid C \mid CB \mid CF \mid CFB \mid FB$
 $F \rightarrow aFa \mid abaA \mid aba\}$

Druhý, praktickejší postup v úlohe 6.2.

Daná je bezkontextová gramatika $G = (N, T, P, S)$, pričom $N = \{S, A, B, C, D\}$, $T = \{a, b\}$,

$P = \{ \begin{array}{l} S \rightarrow aB \mid BAab \\ A \rightarrow aBa \mid BaC \mid DaBC \\ B \rightarrow CBC \mid CD \\ C \rightarrow \varepsilon \mid aCBa \\ D \rightarrow BC \mid C \mid bBa \end{array} \}$

Zostrojte takú bezkontextovú gramatiku G' , ktorá neobsahuje *epsilon* na pravej strane žiadneho pravidla a platí $L(G) = L(G')$. Použite konštrukciu z prednášky.

RIEŠENIE:

Použijeme trochu iný prístup ako v úlohe 6.1. a odstránime všetky naraz:

$H_0 = \{C\}$
 $H_1 = \{D\} \cup H_0$
 $H_2 = \{B\} \cup H_1 = H = \{B, C, D\}$

Teraz každé pravidlo, kde sa vpravo vyskytne jeden z neterminálov z H skopírujeme a neterminál v skopirovanom pravidle nahradíme ε , čo je ekvivalentné jeho vymazaniu.

Ak sa neterminál z H v pravidle vyskytne viacej, musíme vytvoriť každú možnú kombináciu ich miznutia, takže potrebujeme $(n!)-1$ kópií, v ktorých mažeme $(-1$ lebo originál ostáva, a nepočítame duplicitu, vtedy ich stačí menej).

Gramatika G' bude :

$G' = (N', T', P', S')$, kde $N' = N$, $T' = T$, $S' = S$ a

$P' = \{ \begin{array}{l} S \rightarrow aB \mid BAa\bar{a}b \mid a (=aB, \text{ kde za } B \text{ sme nahradili } \varepsilon) \mid Aa\bar{a}b \\ A \rightarrow aBa \mid BaC \mid DaBC \mid aa \mid Ba \mid aC \mid a \mid aBC \mid DaC \mid DaB \mid Da \mid aB \mid aC \\ B \rightarrow CBC \mid CD \mid CB \mid D \mid C \mid CC \mid BC \mid B \\ C \rightarrow aCBa \mid aBa \mid aCa \mid aa \\ D \rightarrow BC \mid C \mid bBa \mid B \mid ba \end{array} \}$

Daná je bezkontextová gramatika $G = (N, T, P, S)$, pričom $N = \{S, A, B, C, D\}$, $T = \{a, b, c\}$,

$P = \{$
 $S \rightarrow A \mid BC \mid D$
 $A \rightarrow B \mid C \mid aD$
 $B \rightarrow aDD \mid SDS$
 $C \rightarrow S \mid aCB$
 $D \rightarrow Dc \mid b \}$

Zostrojte bezkontextovú gramatiku G' takú, že $L(G) = L(G')$ a G' neobsahuje chain rules.
 (Chain rule je pravidlo typu $F \rightarrow P$, kde F, P sú neterminály.)

Riešenie : Potrebujeme odstrániť viaceré pravidlá, podeme postupovať postupným odstraňovaním nepohodlných pravidiel:

Odstránime pravidlo $A \rightarrow B$

$P_1 = \{$
 $S \rightarrow A \mid BC \mid D \mid B$
 $A \rightarrow C \mid aD$
 $B \rightarrow aDD \mid SDS$
 $C \rightarrow S \mid aCB$
 $D \rightarrow Dc \mid b \}$

Odstránime pravidlo $A \rightarrow C$

$P_2 = \{$
 $S \rightarrow A \mid BC \mid D \mid B \mid C$
 $A \rightarrow aD$
 $B \rightarrow aDD \mid SDS$
 $C \rightarrow S \mid aCB$
 $D \rightarrow Dc \mid b \}$

Nahradíme všetky A dvojicou aD , lebo iná možnosť prechodu z A nie je

$P_3 = \{$
 $S \rightarrow aD \mid BC \mid D \mid B \mid C$
 $B \rightarrow aDD \mid SDS$
 $C \rightarrow S \mid aCB$
 $D \rightarrow Dc \mid b \}$

Odstránime pravidlo $C \rightarrow S$

$P_4 = \{$
 $S \rightarrow aD \mid BC \mid D \mid B \mid C \mid BS \mid S$
 $B \rightarrow aDD \mid SDS$
 $C \rightarrow aCB \mid aSB$
 $D \rightarrow Dc \mid b \}$

Pravidlo $S \rightarrow S$ môžeme vynechať, nijako neovplyvní generovaný jazyk, odstránime pravidlo $S \rightarrow C$

$P_5 = \{$
 $S \rightarrow aD \mid BC \mid D \mid B \mid BS \mid BC$
 $B \rightarrow aDD \mid SDS \mid CDS \mid SDC \mid CDC$
 $C \rightarrow aCB \mid aSB \mid aCB$
 $D \rightarrow Dc \mid b \}$

Odstránime pravidlo $S \rightarrow B$

$$P_6 = \{ \begin{array}{l} S \rightarrow aD \mid BC \mid D \mid BS \mid BC \mid BB \\ B \rightarrow aDD \mid SDS \mid CDS \mid SDC \mid CDC \mid BDS \mid SDB \mid BDB \mid CDB \mid BDC \\ C \rightarrow aCB \mid aSB \mid aCB \mid aBB \\ D \rightarrow Dc \mid b \end{array} \}$$

Odstránime pravidlo $S \rightarrow D$

$$P = \{ \begin{array}{l} S \rightarrow aD \mid BC \mid BS \mid BC \mid BB \mid BD \\ B \rightarrow aDD \mid SDS \mid CDS \mid SDC \mid CDC \mid BDS \mid SDB \mid BDB \mid CDB \mid BDC \mid DDS \mid \\ SDD \mid DDD \mid CDD \mid DDC \mid BDD \mid DDB \\ C \rightarrow aCB \mid aSB \mid aCB \mid aBB \mid aDB \\ D \rightarrow Dc \mid b \end{array} \}$$

Toto je naša hľadaná gramatika, teda jej pravidlá odvodzovania.

Zostrojte bezkontextovú gramatiku generujúcu jazyk $L = \{w \mid w \text{ patrí do } \{a, b\}^* ; \#_a(w) = \#_b(w)\}$.
Dokážte správnosť svojej konštrukcie.

Riešenie :

Riešenie sa nachádza v skriptách (Formálne jazyky a automaty, poznámky z prednášok, verzia 0.98) na strane 96, takže nevidím dôvod, prečo niečo vymýšľať, kto ich nemá po ruke tak tu je riešenie :

$G = (N, \{a, b\}, P, S)$, $N = \{S, A, B\}$

$P = \{ \quad S \rightarrow \varepsilon \mid aB \mid bA$

$\quad \quad A \rightarrow aS \mid bAA$

$\quad \quad B \rightarrow bS \mid aBB$

$\quad \quad \}$

Dôkaz očakáva záujemcu v spomenutých skriptách.

Zostrojte DKA akceptujúci jazyk $L = \{a^n \mid 2 \mid n \vee 3 \mid n\}$. Dokážte správnosť svojej konštrukcie.

Riešenie :

$A = (K, \Sigma = \{a\}, \delta, q_0, F)$, kde $K = \{q_0, q_1, q_2, q_3, q_4, q_5\}$,
 $F = \{q_0, q_2, q_3, q_4\}$.

Prechodová funkcia má tvar:

$$\delta(q_0, a) = \{q_1\}$$

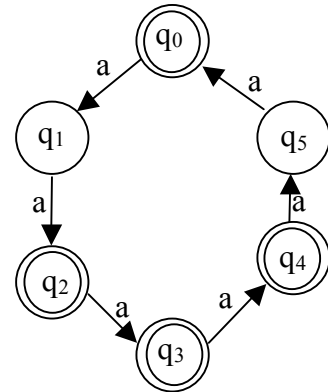
$$\delta(q_1, a) = \{q_2\}$$

$$\delta(q_2, a) = \{q_3\}$$

$$\delta(q_3, a) = \{q_4\}$$

$$\delta(q_4, a) = \{q_5\}$$

$$\delta(q_5, a) = \{q_0\}$$



Dôkaz neformálne:

Index stavu q je rovnaký ako zvyšok po delení počtu a číslom 6. Keď je zvyšok 0, 2, 4, tak je číslo deliteľné 2-mi, ak je 0 alebo 3, tak je deliteľný 3-mi, preto ak na konci výpočtu automatu som v niektorom z uvedených stavov, tak je slovo z L a ak nie, tak zvyšok po delení 2 ani 3 nie je nula a to je ekvivalentné tomu, že slovo ni je z L .

Zostrojte NKA akceptujúci jazyk $L = \{a^n \mid 2 \mid n \vee 3 \mid n\}$. Dokážte správnosť svojej konštrukcie.

Riešenie :

$A = (K, \Sigma = \{a\}, \delta, q_0, F)$, kde $K = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, $F = \{q_0, q_p, q_t\}$.

Prechodová funkcia má tvar:

$$\delta(q_0, a) = \{q_{p+1}\}$$

$$\delta(q_0, a) = \{q_{t+1}\}$$

$$\delta(q_{p+1}, a) = \{q_p\}$$

$$\delta(q_p, a) = \{q_{p+1}\}$$

$$\delta(q_{t+1}, a) = \{q_{t+2}\}$$

$$\delta(q_{t+2}, a) = \{q_t\}$$

$$\delta(q_t, a) = \{q_{t+1}\}$$

Tento automat sa na začiatku rozvetví na dve časti, jedna akceptuje slovo s párnym počtom a , druhá slovo s počtom a deliteľným 3-mi.

Poznámka : Dal by sa použiť aj automat z 1. časti, tento je ale explicitne NKA.

Zostrojte NKA akceptujúci jazyk $L = \{w \mid w \text{ patrí do } \{a, b\}^*, w \text{ obsahuje podslovo } aba \text{ alebo } w \text{ obsahuje podslovo } bab\}$. Dokážte správnosť svojej konštrukcie. Zamyslite sa, ako by sa dal skonštruovať DKA akceptujúci L .

Riešenie : Inak povedané, automat akceptuje $\{a,b\}^*aba\{a,b\}^* \cup \{a,b\}^*bab\{a,b\}^*$.

Takto to môžeme rozdeliť automat na tri časti, 1. a 3. akceptujú $\{a,b\}^*$, 2. akceptuje aba a bab .

$A = (K, \Sigma = \{a,b\}, \delta, q_0, F)$, kde $K = \{q_0, q_{1a}, q_{1ab}, q_{1b}, q_{1ba}, q_2\}$,
 $F = \{q_2\}$.

Prechodová funkcia má tvar:

$$\delta(q_0, a) = \{q_0, q_{1a}\}$$

$$\delta(q_0, b) = \{q_0, q_{1b}\}$$

$$\delta(q_{1a}, b) = \{q_{1ab}\}$$

$$\delta(q_{1b}, a) = \{q_{1ba}\}$$

$$\delta(q_{1ab}, a) = \{q_2\}$$

$$\delta(q_{1ba}, b) = \{q_2\}$$

$$\delta(q_2, a) = \{q_2\}$$

$$\delta(q_2, b) = \{q_2\}$$

Dôkaz je tuze triviálny, jak by řekl akademik kořínek a po něm akademik Legéň ☺

Aby sme došli do akceptačného stavu, musíme prejsť cez $q_0, a \rightarrow q_{1a}, b \rightarrow q_{1ab}, a \rightarrow q_2$ alebo cez $q_0, b \rightarrow q_{1b}, a \rightarrow q_{1ba}, b \rightarrow q_2$, čo mi zaručuje, že akceptujem iba slovo obsahujúce buď aba alebo bab , ktoré je z L .

Ak máme slovo z L a chceme ho akceptovať, stačí sa pozrieť, kde sa vyskutoje podslovo, ktoré tam musí byť, nech je na pozícii s indexom i , potom stačí $i-1$ krát použiť pravidlá z q_0 na q_0 , potom akceptovať podslovo a potom zvyšok slova na stave q_2 , ktorý je akceptačným, takže slovo mi automat akceptuje.

Zamyslenie : štandardnou konštrukciou sa dá previesť na DKA.

Zostrojte NKA akceptujúci jazyk $L = \{w \mid w \text{ patrí do } \{a, b\}^*, w \text{ obsahuje podslovo } aba \text{ a zároveň } w \text{ obsahuje podslovo } bab\}$. Dokážte správnosť svojej konštrukcie. Zamyslite sa, ako by sa dal skonštruovať DKA akceptujúci L .

Riešenie: Na rozdiel od úlohy 7.3 (v nej je miesto zároveň alebo), kde stačilo umiestniť akceptáciu podslova vedľa seba, tj. aby automat akceptoval aspoň jedno podslovo, v našej úlohe je potreba ich dať za seba, v oboch kombináciách umiestnenia, tj. musím akceptovať jazyk pozostávajúci zo slov

$\{a, b\}^* aba \{a, b\}^* bab \{a, b\}^*$

alebo

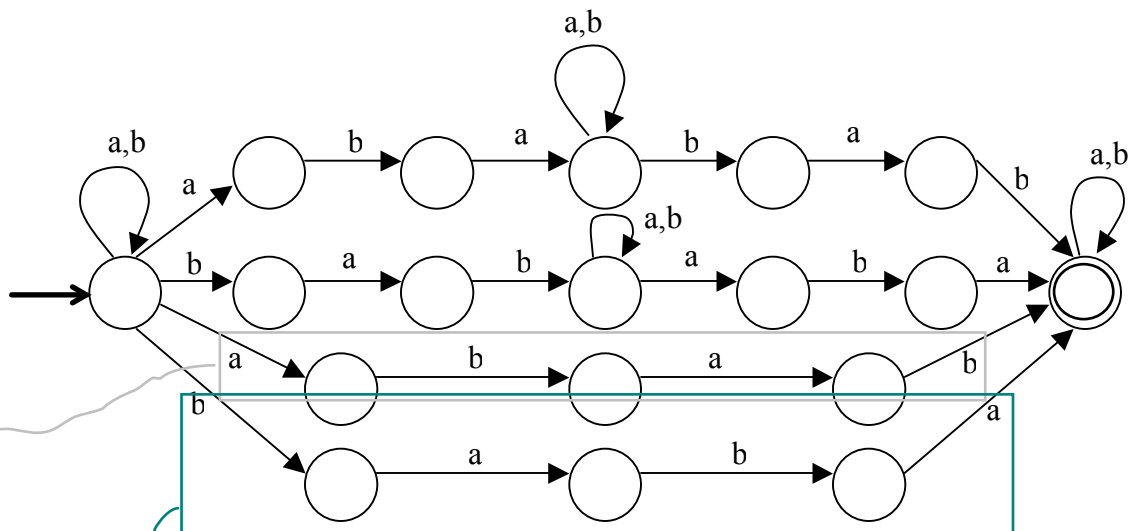
$\{a, b\}^* bab \{a, b\}^* aba \{a, b\}^*$

Ale! Vzniká tu problém, ak sa slová bab a aba prekrývajú, tj. ak ako podreťazec je slovo abab alebo baba. Tieto nepatria ani do jednej množiny, takže automat zostrojený tak aby akceptoval uvedené množiny by akceptoval iba podmnožinu. Musí preto akceptovať aj množiny

$\{a, b\}^* baba \{a, b\}^*$

$\{a, b\}^* abab \{a, b\}^*$

Takže graficky bude vyzeráť nejako takto:



Poznámka : vynechal som označenie stavov, stačilo by ich ľubovoľne pomenovať.

Dôkaz, že automat akceptuje iba slová z L je pomerne jednoduchý, aby prešiel z prvého do posledného stavu, musel by použiť jeden zo štyroch prechodov, v každom z nich sa nachádza slovo požadovaného podtvary.

Opačný dôkaz je trochu ťažší. Postup by bol asi taký, že keď mám slovo z L , tak je buď typu že najskôr ide bab a potom aba alebo opačne (súčasne samozrejme nemôžu). Ak mám prvý typ, tak podslovo bab je na pozícii povedzme i . Takže najskôr akceptujem prvých $i-1$ znakov bez prechodu na ďalší stav. Potom niekde ďalej sa musí nachádzať podslovo aba, nemôže byť na pozícii i , môže byť na $i+1$, $i+2$ byť nemôže, $i+3$ a viac byť môže. Oba prípady mi automat dokáže akceptovať. A tak isto sa dá dokázať pre opačné poradie podslov.

Daný je nedeterministický konečný automat $A = (K, \Sigma, \delta, q_0, F)$, kde $K = \{q_0, q_1, q_2, q_3, q_4\}$, $S = \{a, b\}$ a $F = \{q_4\}$.

Delta funkcia je daná nasledovne:

$$\delta(q_0, a) = \{q_0, q_1\}$$

$$\delta(q_0, b) = \{q_0\}$$

$$\delta(q_1, a) = \{q_3, q_4\}$$

$$\delta(q_1, b) = \{q_2\}$$

$$\delta(q_2, a) = \{q_3\}$$

$$\delta(q_2, b) = \{q_1, q_2\}$$

$$\delta(q_3, a) = \{q_2\}$$

$$\delta(q_3, b) = \{q_2\}$$

$$\delta(q_4, b) = \{q_3\}$$

Zostrojte ekvivalentný deterministický konečný automat.

Riešenie:

Automat $A' = (K', \Sigma, \delta', q_0, F')$, kde $F' = \{q_{0134}, q_{01234}\}$, ostatné sa dá určiť z δ' .

$$\delta'(q_0, a) = \{q_{01}\}$$

$$\delta'(q_0, b) = \{q_0\}$$

$$\delta'(q_{01}, a) = \{q_{0134}\}$$

$$\delta'(q_{01}, b) = \{q_{02}\}$$

Stav q	a	b
0	0,1	0

$$\delta'(q_{0134}, a) = \{q_{01234}\}$$

$$\delta'(q_{0134}, b) = \{q_{023}\}$$

$$\delta'(q_{02}, a) = \{q_{013}\}$$

$$\delta'(q_{02}, b) = \{q_{012}\}$$

Stav q	a	b
0	0,1	0
1	3,4	2
01	0134	02

$$\delta'(q_{01234}, a) = \{q_{01234}\}$$

$$\delta'(q_{01234}, b) = \{q_{0123}\}$$

$$\delta'(q_{023}, a) = \{q_{0123}\}$$

$$\delta'(q_{023}, b) = \{q_{012}\}$$

$$\delta'(q_{013}, a) = \{q_{01234}\}$$

$$\delta'(q_{013}, b) = \{q_{02}\}$$

Stav q	a	b
0	0,1	0
1	3,4	2
3	2	2
4	-	3
0134	01234	023

$$\delta'(q_{012}, a) = \{q_{0134}\}$$

$$\delta'(q_{012}, b) = \{q_{012}\}$$

$$\delta'(q_{0123}, a) = \{q_{0134}\}$$

$$\delta'(q_{0123}, b) = \{q_{012}\}$$

Postup odvodovania je zrejmý z tabuliek pre q_0 , q_{01} a q_{0134} .

ZADANIE:

Daný je **NKA** $A = (K, \Sigma, \delta, q_0, F)$, kde:

$K = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{a, b\}$ a $F = \{q_4\}$.

δ funkcia je daná nasledovne:

$\delta(q_0, a) = \{q_0, q_1\}$

$\delta(q_0, b) = \{q_0\}$

$\delta(q_1, a) = \{q_3, q_4\}$

$\delta(q_1, b) = \{q_2\}$

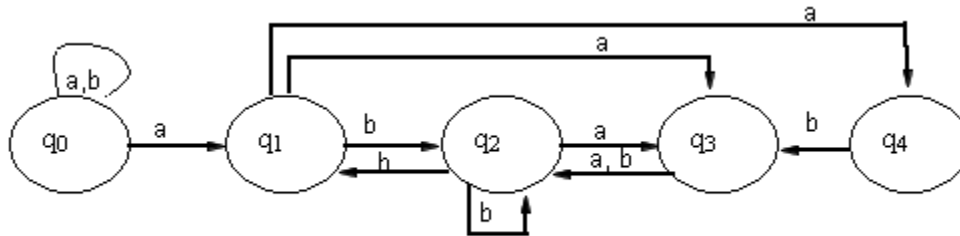
$\delta(q_2, a) = \{q_3\}$

$\delta(q_2, b) = \{q_1, q_2\}$

$\delta(q_3, a) = \{q_2\}$

$\delta(q_3, b) = \{q_2\}$

$\delta(q_4, b) = \{q_3\}$



Zostrojte ekvivalentný **DKA**. //(zrus prechody na ϵ)

RIEŠENIE:

DKA Automat $A' = (K', \Sigma, \delta', q_0, F')$, kde $F' = \{q_{0134}, q_{01234}\}$, ostatné sa dá určiť z δ' :

$\delta'(q_0, a) = \{q_{01}\}$

$\delta'(q_0, b) = \{q_0\}$

$\delta'(q_{01}, a) = \{q_{0134}\}$

$\delta'(q_{01}, b) = \{q_{02}\}$

Stav q	a	b
0	0,1	0

$\delta'(q_{0134}, a) = \{q_{01234}\}$

$\delta'(q_{0134}, b) = \{q_{023}\}$

Stav q	a	b
0	0,1	0
1	3,4	2
01	0134	02

$\delta'(q_{02}, a) = \{q_{013}\}$

$\delta'(q_{02}, b) = \{q_{012}\}$

$\delta'(q_{01234}, a) = \{q_{01234}\}$

$\delta'(q_{01234}, b) = \{q_{0123}\}$

Stav q	a	b
0	0,1	0
1	3,4	2
3	2	2
4	-	3
0134	01234	023

$$\delta'(q_{023}, a) = \{q_{0123}\}$$

$$\delta'(q_{023}, b) = \{q_{012}\}$$

$$\delta'(q_{013}, a) = \{q_{01234}\}$$

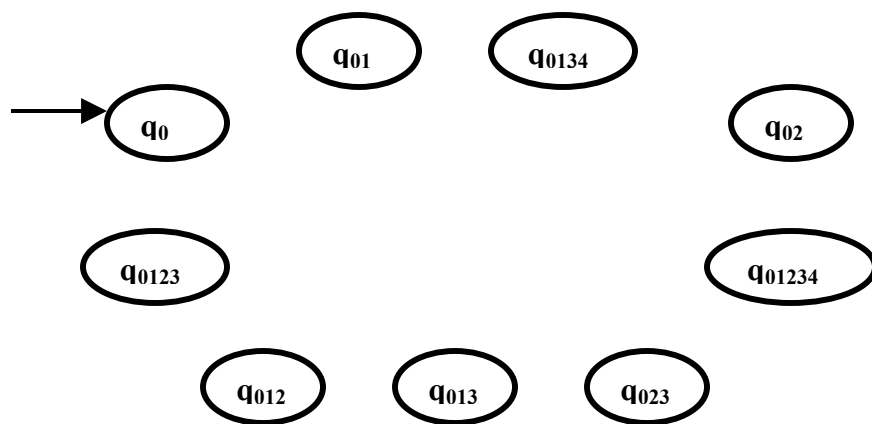
$$\delta'(q_{013}, b) = \{q_{02}\}$$

$$\delta'(q_{012}, a) = \{q_{0134}\}$$

$$\delta'(q_{012}, b) = \{q_{012}\}$$

$$\delta'(q_{0123}, a) = \{q_{0134}\}$$

$$\delta'(q_{0123}, b) = \{q_{012}\}$$



Postup odvod'ovania je zrejmý z tabuliek pre q_0 , q_{01} a q_{0134} .

Daný je nedeterministický konečný automat $A = (K, \{a, b\}, \delta, q_0, F)$, kde
 $K = \{q_0, q_1, q_2, q_3, q_4\}$ a $F = \{q_0, q_4\}$
delta funkcia je daná nasledovne:

$$\delta(q_0, a) = \{q_1, q_3\}$$

$$\delta(q_0, b) = \{q_2, q_4\}$$

$$\delta(q_1, a) = \{q_2\}$$

$$\delta(q_2, b) = \{q_3\}$$

$$\delta(q_2, \varepsilon) = \{q_4\}$$

$$\delta(q_3, a) = \{q_0\}$$

$$\delta(q_3, b) = \{q_1, q_4\}$$

$$\delta(q_4, a) = \{q_0\}$$

Zostrojte ekvivalentný deterministický konečný automat.

Riešenie :

Najskôr odstránime ε - prechod zrušením daného pravidla a pridaním stavu q_4 do tých pravidiel, kde sa prechádza na q_2 , tj. $\delta(q_1, a) = \{q_2, q_4\}$.

Nový automat A' bude mať prechodovú funkciu δ' podľa tabuľky, okrem riadkov pre stavy 1,2,4, tieto nový automat nepotrebuje, keďže sa do nich "nevie dostať" – nie ako do samotného stavu ako napríklad do stavu 3.

Akceptačné budú stavy :

$$F = \{q_0, q_{24}, q_{024}, q_{14}, q_{013}, q_{234}, q_{01234}, q_{124}, q_{1234}, q_{134}\}$$

Stav q	a	b
0	13	24
1	24	-
2	-	3
3	0	14
4	0	-
13	024	14
24	0	3
024	013	234
14	024	-
013	01234	124
234	0	134
01234	01234	1234
124	024	3
1234	024	134
134	024	14

zeby "prasiatkovy" normalny tvar ?

Dokážte, že nedeterministické konečné automaty s počiatočným stavom, do ktorého neexistuje žiadny delta-prechod a s jediným koncovým stavom, z ktorého neexistuje žiadny delta-prechod, sú normálnym tvarom pre konečné automaty.

Riešenie :

Pomerne triviálna úloha. Potrebujeme dokázať, že trieda jazykov generovaná NKA a požadovaným typom automatu je rovnaká, stačí nám ukázať, že ľubovoľný NKA vieme prerobiť na požadovaný typ a a automat zadaného typu vieme prerobiť na NKA.

Prerobenie zadaného typu na NKA je bez práce, keďže daný typ je už typu NKA.

Opačným smerom, prerobenie zadaného NKA $A = (K, \Sigma, \delta, q_0, F)$ na požadovaný tvar urobíme nasledovne :

1. pridaním dvoch nových stavov, a to $q_b \notin K$, $q_e \notin K$,
2. zmenou prechodovej funkcie δ pridaním pravidla $\delta(q_b, \varepsilon) = \{q_0\}$ a pravidiel $\delta(q_i, \varepsilon) = \{q_e\}$ pre každé $q_i \in F$
3. Zmenou počiatočného stavu na q_b a množiny akceptačných stavov na $\{q_e\}$

Dôkaz, že jazyk akceptovaný pôvodným automatom a novým automatom je ten istý je jednoduchý.

Ak sme mali v pôvodnom akceptačný výpočet, tak v novom prejdeme pomocou ε zo stavu q_b na q_0 , ďalší výpočet sa nezmení až po pôvodný akceptačný stav, keď už vstupné slovo je ε , tam prejdeme pomocou ε zo stavu q_i na q_e , a ten je už akceptačný.

Ak mám v novom automate akceptačný výpočet na nejakom slove, tak automat začína a končí nejakým ε - prechodom, ale tie nemenia vstupné slovo a z druhé a predposledného stavu v novom automate existuje ekvivalentný výpočet v pôvodnom automate.

Daná je regulárna gramatika $G = (N, T, P, \sigma)$. Zostrojte regulárnu gramatiku $G' = (N', T, P', \sigma')$ takú, aby pre jazyk $L(G')$ platilo $L(G') = [L(G)]^k$

Riešenie :

V novej gramatike je zrejmé, že $N' = N$. Pre T' platí $T' = T \cup \{\sigma'\}$.

Použijeme predpoklad, že gramatika je v normálnom tvare a všetky pravidlá sú tvaru $N \rightarrow N$ alebo $N \rightarrow T$ alebo $N \rightarrow \epsilon$

(dôkaz – no, nie celkom ☺, v skriptách (neoficiálnych) na strane 19 ako Lema 4.1.1)

Potrebovali by sme tieto pravidlá otočiť a dosiahnuť opačný smer. Budeme postupovať nasledovným postupom:

U pravidiel typu $N \rightarrow N$ je to jednoduché, ak sme mali pravidlo typu :

$\alpha \rightarrow \beta$ tak ho prepíšeme na $\beta \rightarrow \alpha$.

Podobne sa dá postupovať u pravidiel typu

$\alpha \rightarrow a\beta$, prepíšeme ich na $\beta \rightarrow a\alpha$.

Problém nastáva s koncovými pravidlami typu $N \rightarrow T$ a $N \rightarrow \epsilon$, pretože ani jedno sa nedá triviálne pretočiť. Musíme preto vytvoriť špeciálne množiny ako $[a]$, resp. $[\epsilon]$, ktoré mi označujú množinu neterminálov, pre ktoré existuje odvodenie na a , resp ϵ , tj. $[a] = \{\xi \mid \xi \rightarrow a \in P.\}$

S pomocou týchto množín už môžem zadať zvyšok pravidiel.

Potrebujem definovať pravidlá pre σ' , ako

$\sigma' \rightarrow [\epsilon]$ - otočenie pravidiel typu $N \rightarrow \epsilon$

$\sigma' \rightarrow a[a]$ - iba ak $[a] \neq \emptyset$ - otočenie pravidiel typu $N \rightarrow a$, ak také existuje

Ešte zostáva pridať pravidlo $\sigma \rightarrow \epsilon$, pretože “prvé použité pravidlo” je typu $\epsilon \rightarrow \sigma$.

Takže stručne, z počiatočného N pridáme ľubovoľný možný T a za ním N , ktoré sa mohli vyskytnúť na konci pôvodného odvodenia (slova), resp. prejdeme na N prepísateľný na ϵ - tiež musel byť na konci. Potom pridávam T ktoré sú zviazané s určitým N . Dá sa tiež na to pozerieť, ako keby som v pôvodnom odvodovaní nemazal prepísané N , napr.

$\sigma \rightarrow a\alpha \rightarrow ab\alpha \rightarrow abb\beta \rightarrow abb$
($\sigma \rightarrow a\alpha, \alpha \rightarrow b\alpha, \alpha \rightarrow b\beta, \beta \rightarrow \epsilon$)

sa dá zapísať aj ako $\sigma a\alpha b\alpha b\beta$, kde je vidno, že každý T je obklopený zľava aj sprava T , a to nám umožňuje preklopenie naspäť. Na konci musíme prejsť na σ , lebo z toho sme vyšli v pôvodnom odvodení, v novom je to jediný možný stav, ktorým sa dá ukončiť nové odvodenie, lebo je to jediný neterminál, ktorý má prechod na ϵ (v novej gramatike prechod na T nemôže byť, lebo sme ju tak zadefinovali, aby tam nebol).

1. Daný je nedeterministický konečný automat $A = (K, S, \delta, q_0, F)$, kde $K = \{q_0, q_1, q_2, q_3, q_4\}$, $S = \{a, b\}$ a $F = \{q_1\}$.

Delta funkcia je daná nasledovne:

Zostrojte ekvivalentnú regulárnu gramatiku.

$\delta(q_0, a)$	=	$\{q_0, q_1\}$
$\delta(q_0, b)$	=	$\{q_0\}$
$\delta(q_1, a)$	=	$\{q_3, q_4\}$
$\delta(q_1, b)$	=	$\{q_2\}$
$\delta(q_2, a)$	=	$\{q_3\}$
$\delta(q_2, b)$	=	$\{q_1, q_2\}$
$\delta(q_3, a)$	=	$\{q_2\}$
$\delta(q_3, b)$	=	$\{q_2\}$
$\delta(q_4, b)$	=	$\{q_2\}$

Riešenie : triviálne, iba prepíšeme

$G = (N, T, P, q_0)$,

$N = \{q_0, q_1, q_2, q_3, q_4\}$

$T = \{a, b\}$

$P = \{ \begin{array}{l} q_0 \rightarrow aq_0 \mid aq_1 \mid bq_0 \\ q_1 \rightarrow aq_3 \mid aq_4 \mid bq_2 \\ q_2 \rightarrow aq_3 \mid bq_1 \mid bq_2 \\ q_3 \rightarrow aq_2 \mid bq_2 \\ q_4 \rightarrow bq_3 \mid \varepsilon \end{array} \}$

Je vidno postup, akceptovaný znak = pripísaný, akceptačný stav znamená prechod na ε .

Daný je nedeterministický konečný automat $A = (K, \{a, b\}, \delta, q_0, F)$, kde

$K = \{q_0, q_1, q_2, q_3, q_4\}$ a $F = \{q_3, q_4\}$

Delta funkcia je daná nasledovne:

$$\delta(q_0, a) = \{q_1, q_2\}$$

$$\delta(q_0, b) = \{q_2, q_4\}$$

$$\delta(q_1, a) = \{q_2\}$$

$$\delta(q_2, b) = \{q_3\}$$

$$\delta(q_2, \varepsilon) = \{q_4\}$$

$$\delta(q_3, a) = \{q_3\}$$

$$\delta(q_3, b) = \{q_1, q_4\}$$

$$\delta(q_4, a) = \{q_0\}$$

Riešenie :

$G = (N, T, P, q_0)$, kde

$N = \{q_0, q_1, q_2, q_3, q_4\}$

$T = \{a, b\}$

$P = \{ q_0 \rightarrow aq_1 \mid aq_3 \mid bq_2 \mid bq_4$

$q_1 \rightarrow aq_2$

$q_2 \rightarrow q_4$

$q_3 \rightarrow aq_0 \mid bq_1 \mid bq_4$

$q_4 \rightarrow aq_0 \mid \varepsilon \}$

Nech A je NKA, zostrojte NKA A' taký že $L(A') = (L(A))^R$. (L^R je reverz jazyka L).

Riešenie:

Máme NKA A s určitými pravidlami nad abecedou Σ .

Nech $A = (K, \Sigma, \delta, q_0, F)$, kde $K = \{q_0, \dots, q_n\}$, $F \subseteq K$.

Budeme potrebovať špeciálne množiny, označíme ich :

$$[q_i]^a = \{q \in K \mid q_i \in \delta(q, a)\},$$

tj. množina tých stavov, z ktorých existuje výpočet do stavu q_i pomocou terminálu a .

Môžeme následne skonštruovať automat A' takto : $A' = (K', \Sigma, \delta', q', F')$, kde

$K' = K \cup \{q'\}$, $q' \notin K$, - tj, nejaký nový stav, bude počiatočným

$F' = \{q_0\}$, - akceptačný stav je počiatočný v pôvodnom automate,

$\delta'(q, a) = [q]^a$, $q \in K$, $a \in \Sigma$

$\delta'(q', \varepsilon) = F$

Tj. z nového počiatočného stavu q' prejdeme najskôr na pôvodné akceptačné stavy, potom prechádzam „otočenými“ pravidlami a keď som v pôvodnom počiatočnom stave, tak som v novom akceptačnom stave.

Daná je gramatika $G = (N, T, P, S)$, kde $N = \{S, A, B, C\}$, $T = \{a, b\}$ a

$$P = \{ \begin{array}{l} S \rightarrow aC \mid abA \mid bb \\ A \rightarrow aC \mid baB \mid C \\ B \rightarrow \varepsilon \mid A \\ C \rightarrow ab \mid aS \end{array} \}$$

Zostrojte k nej ekvivalentný konečný automat.

Riešenie :

Je vhodné mať gramatiku v nejakom “normálnom” tvare, tj. aby sa pridával maximálne jeden neterminál (ľahko by sa prevádzalo na automat s pružnou hlavou ktorý vie načítať viacero znakov naraz – ten je pravdaže ekvivalentný konečným automatom).

Tam kde je viacero terminálov bude potreba použiť nejaké špeciálne (navyše) stavy.

Nech $A = (K, \{a, b\}, \delta, q_0, F)$,

pre jednoduchosť použijeme na označenie stavov $A, B, C, S = q_0$. Potom

$K = \{q_0, A, B, C, q_1, q_2, q_3, q_4, q_5\}$

$F = \{q_5\}$

$\delta(q_0, a) = \{C, q_1\}$

$\delta(q_0, b) = \{q_2\}$

$\delta(q_1, b) = \{A\}$

$\delta(q_2, b) = \{q_5\}$

$\delta(A, a) = \{C\}$

$\delta(A, b) = \{q_3\}$

$\delta(q_3, a) = \{B\}$

$\delta(A, \varepsilon) = \{C\}$

$\delta(B, \varepsilon) = \{q_5, A\}$

$\delta(C, a) = \{q_4, q_0\}$

$\delta(q_4, b) = \{q_5\}$

Použili sme ďalšie stavy, napr. q_1 aby sme mohli akceptovať $S \rightarrow abA$, $(S \rightarrow aq_1, q_1 \rightarrow bA)$ a podobne.

Je daný jazyk L nad abecedou $\{a, b\}$:

$$L = \{w \mid \#_a(w) \geq \#_b(w)\}$$

Dokážte, že L nie je regulárny jazyk.

Riešenie : Budeme postupovať sporom a predpokladať regulárnosť jazyka L .

Použijeme pumpovaciu lemu. Vezmeme si slovo $w = b^p a^p$, je zrejmé, že $w \in L$, $|w| > p$, preto podľa lemy $\exists x, u, y \in \Sigma^*$ také, že platí :

1. $w = xuy$
2. $u \neq \varepsilon$
3. $|xu| \leq p$
4. $(\forall i \geq 0) : xu^i y \in L$

Z bodu 2 a 3 dostávame, že $u = b^k$, kde $k \in \{1..p\}$ (zároveň vieme, že $xu = b^l$ pre nejaké l)

Použime bod 4, za i zvolíme 2, dostávame : $xuuy = xub^k y = b^{p+k} a^p$, tj. vložili sme do slova b^k . Ale je zrejmé, že $b^{p+k} a^p \notin L$, lebo $\#_a(w) < \#_b(w)$, čím dostávame spor a tvrdenie, že jazyk L je regulárny je vyvrátené, a teda jazyk L nie je regulárny.

Je daný jazyk L nad abecedou $\{a, b\}$:

$$L = \{w \mid \#_a(w) \geq \#_b(w)\}$$

Zostrojte zásobníkový automat akceptujúci prázdnu pamäťou pre jazyk L .

Riešenie :

$A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

$K = \{q_0\}$ - postačí jeden stav, všetko bude otázkou nasledujúceho symbolu a zásobníka

$\Sigma = \{a, b\}$

$\Gamma = \{Z, a, b\}$

$Z_0 = \{Z\}$

$F = \{\}$ - nepotrebujem žiaden akceptačný stav

$\delta(q_0, \varepsilon, Z) = (q_0, \varepsilon)$ - prázdne slovo akceptujem

$\delta(q_0, a, a) = (q_0, aa)$ - pamätám väčší počet

$\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, b) = (q_0, \varepsilon)$ - párujem

$\delta(q_0, b, a) = (q_0, \varepsilon)$

$\delta(q_0, \varepsilon, a) = (q_0, \varepsilon)$ - ak som mal na konci v zásobníku a , tj. bolo viac a , tak ho môžem vybrať a previesť tak automat do akceptačného stavu (ešte vyberiem Z)

Zostrojte zásobníkový automat, ktorý bude akceptovať akc. stavom jazyk:

$$L = \{ww^R \mid w \text{ patrí do } \{a, b\}^*\}$$

Dokážte správnosť svojej konštrukcie.

Riešenie:

Pri konštrukcii budem vychádzať z poznatku, že mojím cieľom nie je určiť stred slova a tak nejako zostrojiť automat, ale automat, ktorého nejaký výpočet bude slovo akceptovať a nebude akceptovať žiadne iné slovo. Takže budem najskôr akceptovať prvú časť slova a odkladať jeho kópiu do zásobníka a potom v určitom okamžiku sa rozhodnem ho skontrolovať so zvyškom a budem postupne vyberať odložené symboly zo zásobníku.

$A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

$$K = \{q_0, q_1, q_2\}$$

$$\Sigma = \{a, b\}$$

$$\Gamma = \{Z, a, b\}$$

$$Z_0 = \{Z\}$$

$$F = \{q_2\}$$

$$\delta(q_0, a, \varepsilon) = (q_0, a) \quad - \text{ukladám do zásobníka}$$

$$\delta(q_0, b, \varepsilon) = (q_0, b) \quad - \text{ukladám do zásobníka}$$

$$\delta(q_0, a, a) = (q_1, \varepsilon) \quad - \text{začal som porovnávať na symbole a}$$

$$\delta(q_0, b, b) = (q_1, \varepsilon) \quad - \text{začal som porovnávať na symbole b}$$

$$\delta(q_1, a, a) = (q_1, \varepsilon) \quad - \text{párujem}$$

$$\delta(q_1, b, b) = (q_1, \varepsilon) \quad - \text{párujem}$$

$$\delta(q_1, \varepsilon, Z) = (q_2, Z) \quad - \text{v zásobníku nezostalo nič na porovnanie, takisto sme akceptovali celé slovo, tak prechádzame do akceptačného stavu}$$

Dôkaz : $L \subseteq L(A)$

Majme slovo $u = ww \in L$. Potom existuje akceptujúci výpočet :

$$(q_0, ww, Z) \vdash^* (q_1, w, Zw^R) \vdash^* (q_1, \varepsilon, Z) \vdash (q_2, \varepsilon, \varepsilon)$$

V prvom kroku sme používali pravidlá na vkladanie do zásobníku pokiaľ sme tam nevložili celé slovo w , potom sme prešli do stavu q_1 a začali porovnávať zásobník so zvyškom slova, a po prejdení na koniec sme použili posledné pravidlo na prechod do akceptačného stavu.

Dôkaz : $L(A) \subseteq L$

Potrebujeme ukázať, že každé akceptované slovo má požadovaný tvar.

Akceptačný výpočet na ľubovoľnom slove $u = a_1 \dots a_n$ musí obsahovať výpočet takéhoto tvaru :

$$(q_0, a_1 \dots a_n, Z) \vdash (q_0, a_2 \dots a_n, Za_1) \vdash (q_0, a_3 \dots a_n, Za_1a_2) \vdash \dots \vdash (q_0, a_m \dots a_n, Za_1a_2a_3 \dots a_{m-1}) \vdash \\ (q_1, a_{m+1} \dots a_n, Za_1a_2a_3 \dots a_{m-2}) \vdash (q_1, a_{m+2} \dots a_n, Za_1a_2a_3 \dots a_{m-3}) \vdash \dots \vdash (q_1, \varepsilon, Z) \vdash (q_2, Z)$$

Najskôr musíme použiť m -krát pravidlá pre stav q_0 , potom môžeme použiť pravidlá pre stav q_1 , a to práve m -krát a príslušné k použitým v stave q_0 , na koniec musíme použiť pravidlo na prechod do stavu

q_2 , ktorý je akceptačný. V momente prechodu už nemôžem mať žiadne vstupné slovo, lebo v stave q_2 už ho neviem akceptovať, takisto v zásobníku už nemôže byť nič iné ako Z , inak by som nemohol prejsť do stavu q_2 .

Tým je dokázané, že automat akceptuje iba slová jazyka L a preto platí dokazovaná rovnosť.

Označme PDA47 taký zásobníkový automat, ktorý má veľkosť zásobníka obmedzenú na 47 znakov. Zadefinujte takýto automat (napíšte štandardé 4 definície). Rozhodnite, či PDA47 tvoria normálny tvar zásobníkových automatov (t.j. či pre každý zásobníkový automat existuje PDA47 automat, akceptujúci ten istý jazyk). Ak áno, dokažte, ak nie, charakterizujte triedu jazykov, ktorú PDA47 akceptujú.

Riešenie :

Definícia 1 Nedeterministickým zásobníkovým automatom typu PDA47 nazývame 7-ticu

$A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

K je konečná množina stavov,

Σ je abeceda vstupných symbolov,

Γ je abeceda zásobníkových symbolov,

q_0 je počiatočný stav,

Z_0 je počiatočný symbol v zásobníku,

$F \subseteq K$ je množina akceptačných stavov a

$\delta : K \times \{\Sigma \cup \varepsilon\} \times \Gamma \rightarrow 2^{K \times \Sigma^*}_{\text{kon}}$ je prechodová funkcia.

Definícia 2 Konfiguráciou nedeterministického zásobníkového automatu typu PDA47 nazývame trojicu

$$\eta \in K \times \Sigma^* \times \Gamma^*$$

kde $g \subseteq \Gamma^*$ a $|g| \leq 47$.

Definícia 3 Krokom výpočtu nedeterministického zásobníkového automatu A typu PDA47 nazývame reláciu \vdash_A na množine konfigurácií definovanú nasledovne:

$$(q, au, \alpha z) \vdash_A (p, u, \alpha \beta) \Leftrightarrow (p, \beta) \in \delta(q, a, z).$$

Definícia 4

Jazykom akceptovaným nedeterministickým zásobníkovým automatom A typu PDA47 **akceptačným stavom** nazývame množinu

$$L(A) =_{\text{def}} \{w \mid \exists q_F \in F, (q_0, w, Z_0) \vdash_A^* (q_F, \varepsilon, \alpha)\}.$$

Jazykom akceptovaným nedeterministickým zásobníkovým automatom A typu PDA47 **prázdnu pamäťou** nazývame množinu

$$N(A) =_{\text{def}} \{w \mid \exists q, (q_0, w, Z_0) \vdash_A^* (q, \varepsilon, \varepsilon)\}.$$

Otázka, či PDA47 tvoria normálny tvar zásobníkových automatov je relatívne ľahká, odpoveď je nie. Zásobníkové automaty typu PDA47 akceptujú triedu regulárnych, trieda PDA47 je ekvivalentná triede nedeterministických konečných automatov. Zdôvodnenie : keďže máme max. 47 symbolov v zásobníku a abeceda zásobníkových symbolov je tiež konečná, môžeme funkciu zásobníka nahradiť stavmi (konečným počtom stavov), kde pri odvodení nového NKA budem mať navyše stav pre každú konfiguráciu zásobníku (odhadom maximálne $(|\Gamma|+1)^{47}$, čo je konečná hodnota). Potom už iba stačí príslušne modifikovať prechodovú funkciu a dostaneme NKA.

To že PDA47 akceptujú regulárne jazyky je zrejmé, stačí prepísať príslušný NKA do PDA47, čo je triviálne.

Rozhodnite a dokážte, či je jazyk $L = \{a^i \mid i \text{ je prvočíslo}\}$ regulárny.

Riešenie: Jazyk L nie je regulárny.

Dôkaz: Budeme postupovať sporom a predpokladať regularitu jazyka L . Použijeme pumpovaciu lemu.

Majme p z pumpovacej lemy, vezmime si slovo $w = a^j$, kde j je nejaké väčšie prvočíslo. Potom musia

$\exists x, u, y \in \Sigma^*$ také, že platí :

1. $w = xuy$
2. $u \neq \varepsilon$
3. $|xu| \leq p$
4. $(\forall i \geq 0) : xu^i y \in L$

Z toho dostávame že u je nenulová mocnina a , povedzme a^k ($k \leq p$, nepodstatné).

Teraz je otázne, či pre každé i v bode 4 dostaneme prvočíslo, je potreba ukázať, že to tak nie je. Zvoľme za i v bode 4 hodnotu j .

Dostávame, že $a^{j+(j.k)} \in L$, lenže číslo $j+j.k = j(k+1)$ evidentne nie je prvočíslo a preto neplatí pumpovacia lema, čím dostávame spor s regularitou jazyka L .

Rozhodnite a dokážte, či je jazyk $L = \{a^i b^j c^k \mid k \leq \min(i, j)\}$ regulárny.

Riešenie: Jazyk L nie je regulárny.

Dôkaz: Budeme postupovať sporom a predpokladať regularitu jazyka L . Takto definovaný jazyk by spĺňal pumpovaciu lemu. Pomôže nám vedomosť, že ak L je regulárny, tak aj L^R musí byť regulárny (cvičenie 8.8).

$$L^R = \{c^k b^j a^i \mid k \leq \min(i, j)\}$$

Tento už pumpovacej leme neodolá, ukážme:

Majme p z pumpovacej lemy, vezmime si slovo $w = a^p b^p c^p$, potom musia

$\exists x, u, y \in \Sigma^*$ také, že platí :

1. $w = xuy$
2. $u \neq \varepsilon$
3. $|xu| \leq p$
4. $(\forall i \geq 0) : xu^i y \in L$

Z toho dostávame že u je nenulová mocnina c , povedzme c^j ($j \leq p$, nepodstatné), zvolíme za i v bode 4 povedzme 2, dostávame, že $c^{p+2j} b^p a^p \in L$, ale to je zrejme spor, takže jazyk L neôže byť regulárny.

Rozhodnite a dokážte, či je jazyk $L = \{a^i b^j c^k \mid k \neq i+j\}$ regulárny.

Riešenie: Jazyk L nie je regulárny.

Dôkaz: Budeme postupovať sporom a predpokladať regularitu jazyka L . Potom musí platiť pumpovacia lema:

Majme p z pumpovacej lemy, vezmeme si slovo $w = a^p c^{p+p!}$, potom musia

$\exists x, u, y \in \Sigma^*$ také, že platí :

1. $w = xuy$
2. $u \neq \varepsilon$
3. $|xu| \leq p$
4. $(\forall i \geq 0) : xu^i y \in L$

Z toho dostávame že u je nenulová mocnina a ($u \in \{a\}^*$), povedzme a^j ($j \leq p$, nepodstatné).

Aby platila pumpovacia lema, museli by sme dokázať, že existuje $j \in \{1, \dots, p\}$ také, že pre každé i platí :

$$p+j \cdot i \neq p + p!$$

Uvedenú nerovnosť sa ale nedá pre každé i dokázať, čo je vidno z nerovnice v tvare

$$j \cdot i \neq p!$$

kde vieme, že $1 \leq j \leq p$, a preto $j \mid p!$ (j delí $p!$). Preto ak v bode 4 zvolíme za $i = p! / j$, dostaneme sa do sporu s tvrdením, že $a^{p+p!} c^{p+p!} \in L$.

Dokážte, že ku každému zásobníkovému automatu existuje ekvivalentný, ktorý má $|\Gamma| = 2$.

Riešenie :

Existuje viacero riešení.

1. Množinu stavov zakódujeme do binárnej sústavy s premenlivou dĺžkou. Ak by sme mali očíslované zásobníkové symboly 1 až n , tak pre vloženie 1 vložíme 01, pre 2 vložíme 011 atď. Potom stačí prerobiť pôvodný automat takýmto spôsobom:

Ak sme mali v pôvodnom nejakú takúto prechodovú funkciu

$$\delta(q_i, a, z_k) = (p_j, z_l)$$

tak musíme najskôr zabezpečiť, že pre nový výpočet bude na vrchu zásobníka symbol z_k :

$$\delta'(q_i, a, 1) = (q_{ia1}, \varepsilon), \delta'(q_{ia1}, \varepsilon, 1) = (q_{ia2}, \varepsilon), \dots, \delta'(q_{ia(k-1)}, \varepsilon, 1) = (q_{ik}, \varepsilon)$$

V tomto stave očakávame na vrchu 0 ako ukončenie jedničiek identifikujúcich symbol v zásobníku, ak sa tam nachádza, tak môžem prejsť na ďalší stav.

$$\delta'(q_{iak}, \varepsilon, 0) = (p_j, [z_j]), \text{ kde } [z_j] \text{ je vyjadrenie v tvare, ktorý sme definovali.}$$

2. Ďalšie a asi elegantnejšie riešenie je zadať zásobníkové symboly v binárnej sústave s konštantnou dĺžkou. To znamená, že ak mám počet symbolov v pôvodnej abecede Γ do 2^n , tak mi stačí použiť dĺžku konštantnú dĺžku n (tj. počet 0 a 1 ktoré musím načítať zo zásobníka, aby som vedel identifikovať pôvodný symbol). Pri odvodovaní nového automatu budeme postupovať podobne ako v bode 1. Rozdiel bude v tom, že budeme mať vždy n pomocných stavov pre načítanie symbolu zo zásobníka, a po prejdení týchto n krokov už viem, aký symbol na vrchu zásobníka bol. Ak mám teda napr. $n=3$, a v zásobníku mám 000010111 (čítam sprava), tak navrchu mám $111 = 7$, potom $010 = 2$ a na konci $000 = 0$.

Dôkaz ekvivalencie je vidno z konštrukcie. Snáď len stručne pre druhý automat:

- Všetko, čo akceptuje pôvodný akceptuje aj nový – výpočet bude prebiehať po tých istých „hlavných“ stavoch, len medzi nimi bude vkladať vždy n krokov navyše, ktorými načíta zásobníkový symbol.
- Všetko, čo akceptuje nový akceptuje aj pôvodný - ak mám nejaký výpočet v novom automate, tak sa dá rozdeliť na časti na „pôvodných“ zásobníkových symboloch a na nových. Keď som v časti na nových symboloch, tak neexistujú pravidlá, ktoré by mi umožňovali nedočítať až po nejaký stav, ktorý bol zhodný s pôvodným.

Ešte sa dá ukázať, že v zásobníku bude vždy n – násobok symbolov ako v pôvodnom odvodení, to je vidno. Ak

Toľko v stručnosti dôkaz.

Označme PDA42 taký zásobníkový automat, ktorý má najviac 42 stavov. Rozhodnite, či PDA42 tvoria normálny tvar zásobníkových automatov (t.j. či pre každý zásobníkový automat existuje PDA42 automat, akceptujúci ten istý jazyk). Ak áno, dokážte, ak nie, charakterizujte triedu jazykov, ktorú PDA42 akceptujú.

Riešenie :

Úlohu by sme mohli trochu pozmeniť, ak existuje PDA42, existuje aj PDA1, resp. koľko je najmenší počet stavov, aby platila ekvivalencia tried.

Riešením je, že triedy sú ekvivalentné, teda PDA42 tvoria normálny tvar zásobníkových automatov.

Inklúzia smerom $PDA42 \subseteq ZA$ je triviálna, keďže PDA42 je iba určitým zúžením zásobníkových automatov.

Opačná inklúzia je pomerne ťažšia, treba načrtnúť postup, ako prerobiť ľubovoľný zásobníkový automat na automat typu PDA42. Vzhľadom na problematiku počtu stavov a rozlišovania popíšem postup vytvorenia PDA1, ktorý spĺňa požiadavky PDA42.

V každom kroku musíme mať možnosť určiť z konfigurácie nového automatu konfiguráciu pôvodného automatu. Využijeme toho, že počet stavov zásobníkového automatu je konečný, takisto aj počet písmen abecedy Γ je konečný, môžeme preto zdefinovať novú Γ' ako množinu pripomínajúcu (a počtom rovnakú) množinu $\Gamma \times K$, tj. množinu symbolov tvaru napr. $\{q\alpha \mid q \in \Gamma, \alpha \in K\}$.

Následne stačí vytvoriť novú prechodovú funkciu ako:

$\delta'(q, a, q\alpha) = ()$ kde $() \in \delta(k, a, \alpha)$

Pomocou PDA ukážte, že trieda bezkontextových jazykov je uzavretá na operáciu ``vloženia", t.j. ukážte, že ak L_1 a L_2 sú bezkontextové jazyky, tak aj jazyk

$$L = \{uvw \mid uv \in L_1 \wedge v \in L_2\}$$

je bezkontextový.

Riešenie:

Máme dva PDA akceptujúce jazyky L_1 a L_2 , $L(A_1) = L_1$ a $L(A_2) = L_2$.
Môžeme predpokladať, že oba dva akceptujú slovo stavom.

Nový automat A bude tiež akceptovať stavom. Zostrojíme ho nasledujúcim postupom.

Nový automat si zaslúži nový začiatok, nech to je stav q_b . Ako počiatočný symbol nech je Z_0' . Najskôr si „založíme“ spodok zásobníka automatu A_1 pomocou prechodu

$$\delta(q_b, \epsilon, Z_0') = (q_0, Z_0', Z_0),$$

kde q_0 je počiatočný stav automatu A_1 a Z_0 je počiatočný symbol zásobníka v automate A_1 .

Následný výpočet bude prebiehať ako v automate v A_1 , podľa rovnakých pravidiel.

Keď sa ale dostanem do akceptačného stavu, musím byť schopný prejsť