Ľudovít Hamaj, 2i2

FoJa - Prezentácia, Úloha 6.2

Zadanie
Dokážte, že $\pounds_{DCF}$ nie je uzavretá na zjednotenie.
(Návod: Porozmýšľajte o jazyku $L = \{a^i b^j c^k \mid i = j \ alebo \ j = k\}$)

Najprv si pripomenme potrebné definície:

Definícia 1
Trieda $\pounds_{DCF}$ je trieda jazykov akceptovaných deterministickými zásobníkovými automatmi akceptačným stavom.

Definícia 2
Deterministický zásobníkový automat (DZA) je 7-tica $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$. Základ je ako pri zásobníkových automatoch a platí:
1. $\delta(q, a, Z)$ obsahuje najviac jeden prvok
2. $\delta(q, \varepsilon, Z)$ obsahuje najviac jeden prvok
3. Ak $\delta(q, \varepsilon, Z) \neq \emptyset$, potom $\delta(q, a, Z) = \emptyset \quad \forall a \in \Sigma$

Definícia 3
Jazyk akceptovaný DZA: $L(A) = \{w \mid (q_0, w, Z_0) \vdash_A^* (q, \varepsilon, \gamma), q \in F, \gamma \in \Gamma^*\}$
Ujasnime si zadanie úlohy. Čo máme vlastne dokázať?
Keby bola $\pounds_{DCF}$ uzavretá na zjednotenie, platilo by, že $\forall L_1, L_2 \in \pounds_{DCF} \Rightarrow L_1 \cup L_2 \in \pounds_{DCF}$
Našou úlohou je dokázať, že $\exists L_1, L_2 (L_1, L_2 \in \pounds_{DCF} \ \bigwedge \ L_1 \cup L_2 \notin \pounds_{DCF})$

Pozrime sa bližšie na jazyk L. Je vlastne zjednotením dvoch jazykov

$$L = \{a^i b^j c^k \mid i = j \ alebo \ j = k\} = \{a^i b^j c^k \mid i = j\} \cup \{a^i b^j c^k \mid j = k\} = \{a^n b^n c^k\} \cup \{a^k b^n c^n\} = L_1 \cup L_2$$

pričom $L_1, L_2 \in \pounds_{DCF}$
To ale znamená, že existuje DZA A_1, pre ktorý platí $L(A_1) = L_1$ a DZA A_2 taký, že $L(A_2) = L_2$.
Aby sme dokázali, že $L_1 \cup L_2 \notin \pounds_{DCF}$, ukážeme, že sa nedá zostrojiť DZA A taký, aby
$L(A) = L(A_1) \cup L(A_2)$.

Zostrojme najprv A_1 pre jazyk $L_1 = \{a^n b^n c^k\}$
Automat by si mal v zásobníku zapamätať počet symbolov "a" a ak sedí s počtom symbolov "b", slovo akceptovať, pričom ho nezaujíma počet symbolov "c".

$A_1 = (K_1, \Sigma, \Gamma_1, \delta_1, q_{10}, Z_{10}, F_1)$; $K_1 = \{q_{10}, q_{11}, q_{12}, q_{13}\}$, $\Sigma = \{a, b, c\}$, $\Gamma_1 = \{Z_{10}, Z_{11}\}$, $F_1 = \{q_{10}, q_{13}\}$

$$\begin{array}{ll}
\delta_1(q_{10}, a, Z_{10}) = \{(q_{11}, Z_{10}Z_{11})\} & (1) \\
\delta_1(q_{11}, a, Z_{11}) = \{(q_{11}, Z_{11}Z_{11})\} & (2) \\
\delta_1(q_{11}, b, Z_{11}) = \{(q_{12}, \varepsilon)\} & (3) \\
\delta_1(q_{12}, b, Z_{11}) = \{(q_{12}, \varepsilon)\} & (4) \\
\delta_1(q_{12}, \varepsilon, Z_{10}) = \{(q_{13}, Z_{10})\} & (5) \\
\delta_1(q_{13}, c, Z_{10}) = \{(q_{13}, Z_{10})\} & (6) \\
\delta_1(q_{10}, c, Z_{10}) = \{(q_{10}, Z_{10})\} & (7)
\end{array}$$

Popis δ-funkcie:
Slová z jazyka $\{a^n b^n c^k\}$ sú jedného zo štyroch tvarov:

1. Nech $n = 0, \ k = 0 \ \Rightarrow \ w = \varepsilon$
   Potom výpočet v automate A_1 má tvar $(q_{10}, \varepsilon, Z_{10})$ a slovo w je akceptované.

2. Nech $n > 0, \ k = 0 \ \Rightarrow \ w = a^n b^n$
   Najprv automat použije pravidlo (1) δ-funkcie a prejde do stavu, v ktorom bude akumulovať do zásobníka za každý symbol "a" symbol Z_{11} (2). Keď prečíta symbol "b", odoberie

zo zásobníka jeden symbol (3) a prejde do stavu, v ktorom pri čítaní symbolov "b" postupne vyprázdňuje zásobník (4). Keď "vyhádže" zo zásobníka všetky symboly Z_{11} (na vrchu zásobníka je Z_{10}), prejde epsilonovým krokom do akceptačného stavu (5).

3. Nech $n = 0,\ k > 0\ \Rightarrow\ w = c^k$
   Automat len prejde v akceptačnom stave cez všetky symboly "c" bez zmeny zásobníka (7). Po dočítaní slova je v stave q_{10}, čo je akceptačný stav.

4. Nech $n > 0,\ k > 0\ \Rightarrow\ w = a^n b^n c^k$
   Automat začne ako v kroku 2. a po prejdení do akceptačného stavu q_{13}, prejde cez všetky symboly "c" bez zmeny stavu a zásobníka (6).

Teraz zostrojme DZA A_2 pre jazyk $L_2 = \{a^k b^n c^n\}$
Ak bude slovo začínať nejakými symbolmi "a", automat cez ne prejde bez povšimnutia, prečíta b-čka a zapamätá si ich počet. Ak sa rovná počtu c-čok, akceptuje.

$A_2 = (K_2, \Sigma, \Gamma_2, \delta_2, q_{20}, Z_{20}, F_2)$; $K_2 = \{q_{20}, q_{21}, q_{22}, q_{23}\}$, $\Sigma = \{a, b, c\}$, $\Gamma_2 = \{Z_{20}, Z_{21}\}$, $F_2 = \{q_{20}, q_{23}\}$

$$
\begin{aligned}
\delta_2(q_{20}, a, Z_{20}) &= \{(q_{20}, Z_{20})\} && (1)\\
\delta_2(q_{20}, b, Z_{20}) &= \{(q_{21}, Z_{20}Z_{21})\} && (2)\\
\delta_2(q_{21}, b, Z_{21}) &= \{(q_{21}, Z_{21}Z_{21})\} && (3)\\
\delta_2(q_{21}, c, Z_{21}) &= \{(q_{22}, \varepsilon)\} && (4)\\
\delta_2(q_{22}, c, Z_{21}) &= \{(q_{22}, \varepsilon)\} && (5)\\
\delta_2(q_{22}, \varepsilon, Z_{20}) &= \{(q_{23}, Z_{20})\} && (6)
\end{aligned}
$$

Popis δ-funkcie:
Podobne ako v automate A_1 len prechádza cez a-čka, akumuluje symboly v zásobníku, keď číta b-čka a vyprázdňuje zásobník, keď číta c-čka.

Chceme sa pokúsiť zostrojiť automat A, ktorý by akceptoval aj tie slová, čo akceptuje A_1, aj tie čo akceptuje A_2. Z minulého semestra si pamätáme konštrukciu nedeterministického zásobníkového automatu A', ktorý mal akceptovať zjednotenie dvoch bezkontextových jazykov akceptovaných dvoma nedeterministickými zásobníkovými automatmi A'_1 a A'_2: Bez ujmy na všeobecnosti sme uvažovali, že množiny stavov, zásobníkových symbolov, akceptačných stavov... sú disjunktné. Potom bola konštrukcia jednoduchá. Nech q'_{10} je počiatočný stav automatu A'_1, q'_{20} počiatočný stav automatu A'_2. Automat A' s počiatočným stavom q_0 sa nedeterministicky rozhodol, z ktorého jazyka je slovo na vstupe a podľa toho prešiel epsilonovým krokom buď do q'_{10} alebo do q'_{20} a simuloval príslušn'y automat. Toto však v deterministickom automate urobiť nemôžme (Definícia 2).

Pokúsme sa teda skonštruovať DZA A.
Nech q_0 je počiatočný stav a Z_0 počiatočný symbol v zásobníku, $w_1 \in L_1$ a $w_2 \in L_2$. Aby akceptoval slovo w_2 musí δ-funkcia obsahovat pravidlo, ktorým prejde na symbol "a" do akumulujúceho stavu a v ňom pridáva na a-čka symboly do zásobníka. Ak slovo w_2 obsahuje na začiatku a-čka, je nám jedno, ako sa na nich automat správa, dôležitý pre nás je prechod, keď na vstupe príde prvé b-čko. A tu nastáva problém. Pokiaľ chceme akceptovať slovo w_1, musíme prejsť do stavu q_2, v ktorom, budeme odoberať zo zásobníka symboly po každom prečítaní b-čka, aby sme zistili, či slovo obsahuje rovnaký počet po sebe idúcich a-čok a b-čok, zatiaľ čo na akceptovanie slova w_2 musíme prejsť do akumulujúceho stavu q_3, aby sme si mohli zapamätať počet b-čok a neskôr ho porovnať s počtom c-čok. Tu by naša δ-funkcia musela mať aspoň dva prvky

$$\delta(q_1, b, Z) = \{(q_2, \varepsilon),\ (q_3, ZZ')\} \quad Z, Z' \in \Gamma$$

čo je spor s **Definíciou 2**.

Pozrime sa na problém ešte cez výpočty a vlastnosti konfigurácií vo význačných bodoch. Významné body sú pre nás miesta prechodu, kde automat dočítal symbol jedného druhu a začína s druhým, resp. skončí. Ukážme si najprv, ako vyzerá výpočet slova w_1 v automate A_1:

$$(q_{10}, a^n b^n c^k, Z_{10}) \vdash^* (q_{11}, b^n c^k, Z_{10} Z_{11}^n) \vdash^* (q_{12}, c^k, Z_{10}) \vdash (q_{13}, c^k, Z_{10}) \vdash^* (q_{13}, \varepsilon, Z_{10})$$

Výpočet slova w_2 v automate A_2:

$$(q_{20}, a^k b^n c^n, Z_{20}) \vdash^* (q_{20}, b^n c^n, Z_{20}) \vdash^* (q_{21}, c^n, Z_{20} Z_{21}^n) \vdash^* (q_{22}, \varepsilon, Z_{20}) \vdash (q_{23}, \varepsilon, Z_{20})$$

Keby existoval automat A taký, že $L(A) = L_1 \cup L_2$, musel by existovať akceptačný výpočet oboch slov - w_1, w_2 v tomto automate. Nech prvá časť výpočtu vyzerá nasledovne:

$$(q_0, a^i b^j c^k, Z_0) \vdash^* (q_1, b^j c^k, Z_0 Z_1^i)$$

Po prečítaní všetkých a-čok si automat proste musí v zásobníku pamätať ich počet, aby bol schopný akceptovať slovo w_1. Zároveň ale vôbec nevadí,keď dostane na vstupe w_2, že výpočet nebeží tak ako v A_2. Povieme si jednoducho, že mu vložíme na epsilonový krok do zásobníka špeciálny symbol Z_S a po naakumulovaní nejakých iných symbolov v zásobníku pri čítaí b-čok a následnom vyprázdnení zásobníka čítaním c-čok prejde epsilonovým krokom do akceptačného stavu, ak má na vrchu zásobníka práve tento špeciálny symbol.

$$(q_1, b^j c^k, Z_0 Z_1^i) \vdash (q_2, b^j c^k, Z_0 Z_1^i Z_S) \vdash^* (q_2, c^k, Z_0 Z_1^i Z_S Z_2^j)...$$

Lenže slovo w_1 potrebuje, aby automat nasledujúcim výpočtom vyberal zo zásobníka symboly Z_1 a overil, či sa ich počty rovnajú:

$$(q_1, b^j c^k, Z_0 Z_1^i) \vdash^* (q_2, c^k, Z_0) \quad i = j$$

To znamená, že nedokážeme deterministicky povedať, čo má automat v nasledujúcom kroku robiť $\rightarrow$ neexistuje automat A, pre ktorý by platilo

$$L(A) = L(A_1) \cup L(A_2) \quad L(A_1), L(A_2) \in \pounds_{DCF}$$

...a z toho vyplýva, že

$$\exists L_1, L_2 (L_1, L_2 \in \pounds_{DCF} \bigwedge L_1 \cup L_2 \notin \pounds_{DCF})$$

Tým sme dokázali, že $\pounds_{DCF}$ nie je uzavretá na zjednotenie.