Cvičení z automatů a gramatik - 3

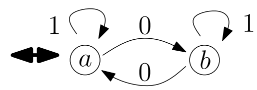
15. a 16. března 2017

Probrané příklady

1. Homomorfismy automatů: definice, zachování přijímaného jazyka.

    (a) Řekneme, že A je *homomorfní na* B, pokud existuje (automatový) homomorfismus $h : Q_A \to Q_B$. Je tato relace reflexivní, symetrická, tranzitivní ?

    (b) Uvažme následující konečný automat C. Nalezněte konečné automaty A, B homomorfní na C (a přitom neizomorfní s C) takové, že A není homomorfní na B a zároveň B není homomorfní na A.

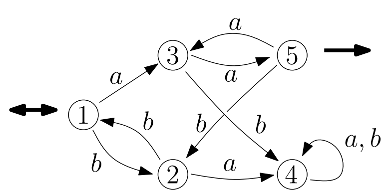


2. Automatová kongruence, podílový automat (faktorstruktura).

    (a) Je stavová ekvivalence po i krocích automatovou kongruencí?

    (b) Uveďte příklad automatové kongruence jiný než je stavová ekvivalence.

3. Redukt: jednoznačnost. Minimalizujte následující konečný automat.



4. Nedeterministický konečný automat: formální definice, výpočet, přijímaný jazyk, interpretace pomocí větvících výpočtů a pomocí uhodnutí přijímacího výpočtu.

    (a) Může být množina počátečních stavů prázdná?

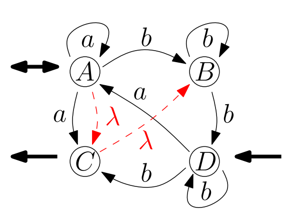
    (b) Můžeme přidat podmínku, že z každého stavu je pro každé písmeno definován aspoň jeden přechod?

    (c) Sestrojte deterministický a nedeterministický automat (s co nejmenší množinou stavů) rozpoznávající jazyk $L = \{w \in \{a, b\}^*; w \text{ končí na } abba\}$.

5. Nedeterministické konečné automaty přijímají právě regulární jazyky.

    (a) Převeďte následující nedeterministický automat (bez λ-přechodů) na deterministický (množinovou konstrukcí).

    (b) Je výsledný deterministický automat (vždy) redukovaný?



6. Rozpoznávání doplňku nedeterministickými konečnými automaty.
    (a) Jaký jazyk dostaneme po přehození koncových/nekoncových stavů u deterministického automatu?
    (b) A u nedeterministického automatu?
    (c) Jak můžeme zadefinovat nový typ nedeterministických konečných automatů (přesněji nový typ přijímaného jazyka), abychom po přehození typu a koncových/nekoncových stavů dostali doplněk původního jazyka?

Domácí úkol

7. Pro každé $n \geq 1$ nalezněte jazyk L_n takový, že každý deterministický automat rozpoznávající L_n má aspoň 2^n stavů, a zároveň L_n lze rozpoznat nedeterministickým automatem s n stavy.

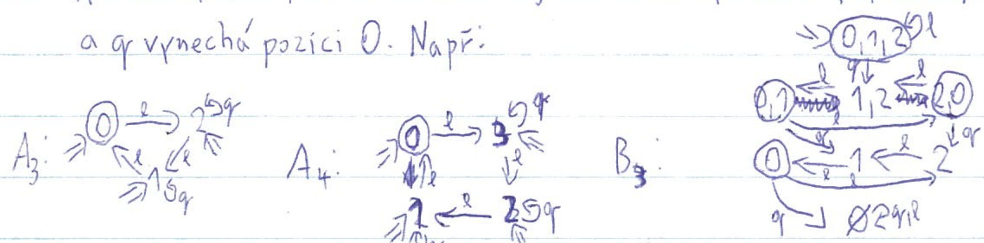
Řešení jen s binární abecedou *(Vít Fojtík)*

Automaty a gramatiky, DÚ3 — Vít Fojtík 1b

Místo jazyka budu uvažovat rovnou o příslušných automatech. Díky větě, že pokud jsou dva DFA ekvivalentní, jsou jejich redukty isomorfní, stačí k příslušnému NFA najít ekvivalentní DFA, který je redukovaný, protože potom je sám svým reduktem, tedy je isomorfní reduktu lib. ekvivalentního DFA, tedy nemá větší počet stavů.

Pro $m \geq 1$ def. NFA A_m takto: $A_m = \langle m, \{l, q\}, \delta_m, m, \{0\} \rangle$, kde $\forall k < m(\delta_m(k, l) = \{k-1 \bmod m\})$, $\forall k \overset{<m}{>} 0(\delta_m(k, q) = \{k\})$, $\delta_m(0, q) = \emptyset$. (Značení: $l(k) = \delta_m(k, l)$, $q(k) = \delta_m(k, q)$). (Moje úmysly se snad objasní dále). Jako hledaný jazyk def. $L_m = L(A_m)$. Nyní pro A_m sestrojím potenční DFA B_m (jehož ekvivalence plyne z konstrukce, resp. cvičení) a ukážu, že je redukovaný.

$B_m = \langle \mathcal{P}(m), \{l, q\}, \gamma_m, m, \{x \in \mathcal{P}(m) \mid 0 \in x\} \rangle$, $\gamma_m(x, l) = \bigcup l'' x$, $\gamma_m(x, q) = \bigcup q'' x$ (opět budu značit $l(x)$, $q(x)$, zmatení snad nehrozí). Intuice: $x \in \mathcal{P}(m)$ si představuji jako označení určitých pozic na „pásce" $0, \dots, m-1$, kde na začátku jsou označeny všechny pozice, l zrotuje pozice doleva a q vynechá pozici 0. Např.:



Dosažitelnost všech stavů: „klesavou indukcí" dle $|x|$, $x \in \mathcal{P}(m)$: $|x| = m \rightsquigarrow x = m \rightsquigarrow x$ je počáteční stav. Buď $|x| = k-1$ a j nejmenší takové, že $j \notin x$. Def. $x' = x \cup \{j\}$ (dosažitelné z IP). Potom $l^{m-j}(q(l^j(x'))) = x$ (Buď $m \in x'$. Pokud $m \neq j$, pak $l^j(m) \neq \{0\} \rightsquigarrow q(l^j(m)) = l^j(m) \rightsquigarrow l^{m-j}(q(l^j(m))) = l^m(m) = \{m\}$. Pokud $m = j$, pak $q(l^j(m)) = \emptyset$).

Redukovanost: Pro spor mějme x, x' ekvivalentní stavy, $x \neq x'$. Buď k nejmenší tž (BÚNO) $k \in x \wedge k \notin x'$. Potom $l^k(x) \in F_{B_m} \wedge l^k(x') \notin F_{B_m}$ (Indukcí: $k = 0$, $y \in F_{B_m} \Leftrightarrow 0 \in y$, $0 \in x \wedge 0 \notin x' \rightsquigarrow x \in F_{B_m} \wedge x' \notin F_{B_m}$. Pro $k+1$: $l(x)$ je ekv. $l(x')$ z def. stavové ekvivalence, navíc $\forall j < k(j \in l(x) \Leftrightarrow j \in l(x'))$ a $k \in l(x)$, $k \notin l(x')$, takže jsou splněny předpoklady a z IP je $l^k(l(x)) \in F_{B_m} \wedge l^k(l(x')) \notin F_{B_m}$.) $\rightsquigarrow$ ↯

Tím je snad hotovo.