ÚVOD DO TI 2011

poznámky k príkladu 7.1

Väčšina z Vás mala správne odpovede na všetky otázky. To, čo Vám však chýbalo boli odôvodnenia (maximálnosti, resp. minimálnosti mohutností) a príklady jazykov, pre ktoré sa dosahuje minimum/maximum.

Keďže všetky časti úlohy boli veľmi podobné ukážeme si riešenie jednej časti, maximálneho počtu prvkov v prieniku. Príkladov sa dalo nájsť viacero, napríklad stačí zvoliť $L_1 = \{a^1, \ldots, a^4\}$ s $L_2 = \{a^1, \ldots, a^7\}$. Prienik bude mať 4 prvky, $a^1, \ldots, a^4$. Dôkazom, že táto hodnota je maximálna, je fakt, že $L_1 \supseteq L_1 \cap L_2$ preto aj $4 = |L_1| \geq |L_1 \cap L_2|$ z čoho vyplýva, že viac ako 4 prvky v prieniku mať nemôžeme.

poznámky k príkladu 7.2

a) Väčšina ľudí nemala problém s návrhom algoritmu. Bolo však treba aspoň pár vetami zdôvodniť, či ich algoritmus naozaj vykoná $O(|w||E|^2)$ operácií.

b) Ľahšia podúloha nebola taká problematická. Niektorí ľudia zabudli do svojho výrazu zahrnúť prázdne slovo. Niektorí zas zabudli na slová tvaru $abc(a + b + c)(a + b + c)^*$.

   Druhá podúloha bola ťažšia, takmer nikto ju nespravil. Tí, ktorí sa ju snažili riešiť, dostali regulárny výraz, ktorý nezahŕňal všetky potrebné slová - na to si treba dať pozor. Dalo sa to riešiť aj cez automat - dal sa skonštruovať 6 stavový automat a z neho sa dal všeobecnou konštrukciou zistiť regulárny výraz. (Ak sa to niekomu zdá veľa vypisovania, tak sa to dá napríklad aj ľahko naprogramovať :))

poznámky k príkladu 7.3

V podúlohe a) väčšina z vás aspoň tušila, čo treba spraviť. Problémom bolo, že ste zabúdali na niektoré veci. Ak spájate dva automaty do jedného s tým, že „zlučujete“ počiatočné stavy, môže sa vám všetko pokaziť prechodom do počiatočného stavu. Existuje síce normálny tvar automatu, ktorý sa nevracia do počiatočného stavu, ale to treba napísať. Iným problémom bolo, že ak vytvoríte nový počiatočný stav a spravíte prechody na tie miesta, kam viedli prechody z pôvodných počiatočných stavov, tak nesmiete zrušiť pôvodné počiatočné stavy, a musíte nový stav prehlásiť za akceptačný, ak $\lambda \in L_1 \cup L_2$.

Podúlohu b) vyriešil správne len jeden z vás. Vzhľadom na to, že konštrukcia funguje presne rovnako ako modulárna konštrukcia pre deterministické automaty[1], je to prinajmenšom poľutovaniahodné. V podstate ste písali dva typy riešení – v prvom ste využívali konštrukciu na komplement vymenením akceptačných a ostatných stavov, v druhom ste zapájali automaty B_1 a B_2 za seba. Komplement však pre nedeterministické automaty nemôžete robiť takto: jednoduchý protipríklad je automat rozpoznávajúci všetky slová obsahujúce $abab$. Tento sa v počiatočnom neakceptačnom stave cyklí na celú abecedu, ak z tohto stavu spravíme akceptačný, bude akceptovať všetky slová nad danou abecedou, nie len tie, ktoré neobsahujú $abab$. Ak zapojíte automaty za seba, dostanete zreťazenie, nie zjednotenie. (Ak sa vstup na páske raz prečíta, nemôže sa čítať znova.)

[1] urobíme kartézsky súčin stavov, akceptačné budú tie, ktoré majú oba „podstavy“ akceptačné