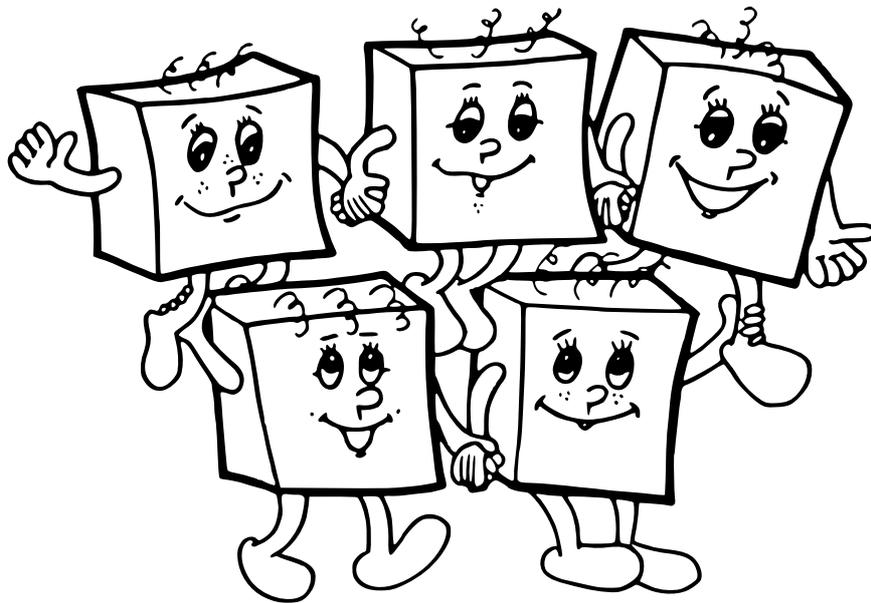


OLYMPIÁDA V INFORMATIKE
NA STREDNÝCH ŠKOLÁCH



dvadsiaty tretí ročník
školský rok 2007/08

zadania celoštátneho kola
kategória A

1. súťažný deň

- **Olympiáda v informatike** je od školského roku 2006/07 samostatnou súťažou. Predchádzajúcich 21 ročníkov tejto súťaže prebiehalo pod názvom **Matematická olympiáda, kategória P** (programovanie).
- Oficiálnu **webstránku** súťaže nájdete na <http://www.ksp.sk/oi/>.



Informácie a pravidlá

Priebeh celoštátneho kola

Celoštátne kolo 23. ročníka Olympiády v informatike, kategórie A, sa koná v dňoch 20.–23. apríla 2008. Na riešenie úloh prvého, teoretického dňa majú súťažiaci 4,5 hodiny čistého času. Rôzne úlohy riešia súťažiaci na samostatné listy papiera. Akékoľvek pomôcky okrem písacích potrieb (napr. knihy, výpisy programov, kalkulačky) sú zakázané.

Čo má obsahovať riešenie úlohy?

Pokiaľ nie je v zadaní povedané ináč, základným kritériom hodnotenia riešenia je správnosť a efektívnosť navrhnutého algoritmu. Hodnotí sa aj kvalita popisu riešenia a zdôvodnenie tvrdení o jeho správnosti a efektívnosti.

Súčasťou riešenia by mali teda byť:

- **Popis riešenia**, to znamená slovný popis použitého algoritmu, argumenty zdôvodňujúce jeho správnosť (prípadne dôkaz správnosti algoritmu), diskusiu o efektívnosti vášho riešenia (časová a pamäťová zložitosť). Slovný popis riešenia musí byť jasný a zrozumiteľný i bez nahliadnutia do samotného zápisu algoritmu (do programu).
- **Program**. V úlohách **A-III-1** a **A-III-2** treba uviesť dostatočne podrobný zápis algoritmu, najlepšie v tvare zdrojového textu najdôležitejších častí programu v jazyku Pascal alebo C/C++. Zo zápisu môžete vynechať jednoduché operácie ako vstupy, výstupy, implementáciu jednoduchých matematických vzťahov a pod.

V prípade, že používate vo svojom programovacom jazyku knižnice, ktoré obsahujú implementované dátové štruktúry a algoritmy (napr. STL pre C++), v popise algoritmu vysvetlite, ako sa časti programu, ktoré obsahujú volania knižnice, dajú implementovať bez nej bez zhoršenia časovej zložitosti.

V úlohe **A-III-3** je namiesto klasického programu potrebné zostrojiť príslušné prekladacie stroje.



Zadania prvého súťažného dňa

A-III-1 Najväčšia horúčava

„Tento rok bolo strašne horúco! V Hurbanove namerali dokonca rovných štyridsať sedem stupňov! Taká horúčava u nás ešte jakživ nebola!“ nechal sa počuť Tadeáš.

„Ále čoby,“ odvetil Kleofáš, „veď predsa pred trinástimi rokmi bolo vonku ešte oveľa väčšie peklo.“

„Tak potom je toto najväčšie teplo odvtedy.“ zahlásil Tadeáš, no Kleofáš ho opäť sklamal: „Aleba. Nemáš pravdu, aj pred šiestimi rokmi bolo teplejšie.“

Súťažná úloha

Dané sú údaje tvaru „v roku r bola najvyššia nameraná teplota t_r “.

Ďalej sú dané výroky tvaru „v roku q_1 bolo najväčšie teplo od roku q_2 “.

Vašou úlohou bude o každom z výrokov rozhodnúť, či je pravdivý, nepravdivý, alebo otázný.

Výrok je pravdivý, ak platí:

– poznáme teploty pre všetky roky od q_2 po q_1 , vrátane

– v roku q_2 bolo aspoň tak teplo ako v roku q_1 (teda $t_{q_2} \geq t_{q_1}$)

– pre každý rok r medzi q_2 a q_1 platí, že v ňom bolo chladnejšie ako v q_1 , a teda aj chladnejšie ako v q_2 (teda $\forall r \in \{q_2 + 1, \dots, q_1 - 1\}; t_r < t_{q_1}$)

Výrok je otázný, ak pre niektoré roky medzi q_2 a q_1 nie je známa nameraná teplota, ale vieme tieto teploty doplniť tak, aby sa výrok stal pravdivým.

V ostatných prípadoch je výrok nepravdivý.

Formát vstupu

Prvá časť vstupu, ktorá popisuje namerané teploty, začína celým číslom N , udávajúcim počet rokov, pre ktoré vieme nameranú teplotu. Nasleduje N dvojíc celých čísel $r_i t_{r_i}$. Môžete predpokladať, že údaje sú utriedené podľa čísla roku, teda pre každé i platí $r_i < r_{i+1}$.

Druhá časť vstupu, ktorá popisuje výroky, začína číslom M , udávajúcim počet výrokov. Nasleduje M dvojíc čísel $q_1 q_2$.

Môžete predpokladať, že $0 \leq N, M \leq 100\,000$ a $0 \leq r_i, t_{r_i}, q_i \leq 1\,000\,000\,000$. Teploty, ktoré nie sú udané na vstupe, môžu nadobúdať aj neceločíselné hodnoty, aj hodnoty mimo uvedeného rozsahu. Riešenie, ktoré funguje za predpokladu $0 \leq N, M \leq 1\,000$, získa aspoň 5 bodov.

Formát výstupu

Pre každý výrok vypíšte jeden riadok, a v ňom jeho pravdivostnú hodnotu.

Príklad

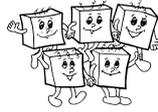
vstup

```
5
2003 49
2004 59
2005 28
2006 38
2008 47
4
2006 2003
2006 2004
2008 2004
2008 2006
```

výstup

```
nepravdivy
pravdivy
otazny
nepravdivy
```

*Tretí výrok môže byť pravdivý napr. ak $t_{2007} = 11$.
Štvrtý výrok nemôže byť pravdivý, lebo $t_{2008} > t_{2006}$.*



A-III-2 Tobogany

V Nalomenej Trieske otvorili včera nový vodný zábavný park. Hlavnou atrakciou je obrovská sieť toboganov.

V sieti je N bazénov, ktoré sa nachádzajú v rôznych výškach nad zemským povrchom. Jeden z nich je horný (tam jazda toboganmi začína) a jeden je dolný (tam každá jazda končí).

Bazény sú troch rôznych tvarov: štvorcový, šesťuholníkový a osemuholníkový. Bazény rovnakého tvaru sú navzájom nerozlišiteľné. Z každého bazénu okrem dolného vedú niekam dodola práve tri tobogany, ktoré sú označené číslami 1, 2 a 3.

Jazda na atrakcii vyzerá tak, že začneme v hornom bazéne a postupne si v každom bazéne, kde sa ocitneme, vyberieme jeden z troch toboganov a pustíme sa ním ďalej dole. Takto pokračujeme, až kým sa nedostaneme do dolného bazénu. Sieť toboganov je navrhnutá tak, že bez ohľadu na to, ako sa budeme rozhodovať, do dolného bazénu sa časom určite dostaneme.

Do každého bazénu môže ústiť ľubovoľne veľa toboganov. Tie sa však spájajú ešte pred tým, ako vás vypľujú do bazéna, takže tento počet neviete zistiť. Jediné, čo na konci jazdy viete, sú čísla toboganov, ktoré ste si vybrali, a tvary bazénov, cez ktoré ste išli.

Zábavný park má pri otvorení veľkú akciu: Kto prvý nakreslí správne mapu celej atrakcie, dostane ročné vstupné zdarma. Malý Ľuboško si zaumienil, že cenu vyhrá. Celý včerajší deň jazdil na toboganoch, až nakoniec nakreslil mapu, ktorá všetkým jeho jazdám zodpovedala. No keď ju skúsil odovzdať, dozvedel sa, že síce má na nej zakreslené všetky bazény, ale bohužiaľ mnohé z nich viackrát.

Súťažná úloha

Na vstupe je daná mapa siete toboganov a bazénov, ktorú nakreslil Ľuboško. Viete, že správna mapa je od Ľuboškovej nerozlišiteľná. To znamená, že ak pôjdete po jednej z máp a budete si voliť čísla toboganov ako len chcete, tak v druhej mape bude vždy taká cesta existovať tiež, a pôjde cez bazény rovnakých tvarov.

Tiež viete, že počet bazénov na správnej mape je najmenší možný. Napíšte program, ktorý zistí, koľko bazénov je na správnej mape.

Formát vstupu

Prvý riadok vstupu obsahuje jediné celé číslo N ($2 \leq N \leq 5000$) udávajúce počet bazénov na Ľuboškovej mape, vrátane horného a dolného. Bazény sú očíslované od 1 po N tak, že bazén s vyšším číslom je bližšie pri zemi. Teda špeciálne bazén 1 je horný a bazén N je dolný bazén.

Nasleduje $N - 1$ riadkov, jeden pre každý bazén okrem dolného. Každý riadok obsahuje štyri záznamy: začína číslom t_i ($t_i \in \{4, 6, 8\}$) udávajúcim tvar i -teho bazénu. Nasledujú tri čísla bazénov, do ktorých vedú z tohto bazéna tobogany s číslami 1, 2 a 3.

Formát výstupu

Výstup má obsahovať jediné číslo – počet bazénov na správnej, najmenej možnej mape.

Príklady

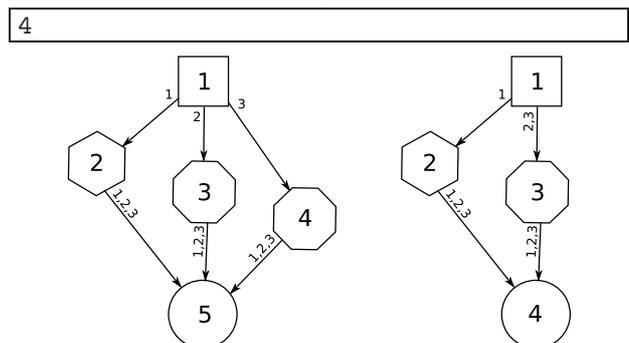
vstup

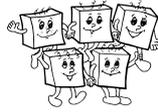
5
4 2 3 4
6 5 5 5
8 5 5 5
8 5 5 5

Zadaná mapa je znázornená na obrázku vľavo. (Všimnite si, že na tvare dolného bazéna nezáleží, preto nie je ani udaný na vstupe. V obrázku ho značíme kruhom.)

Najmenšia mapa, ktorá je nerozoznatelná od zadanej, je na obrázku vpravo.

output





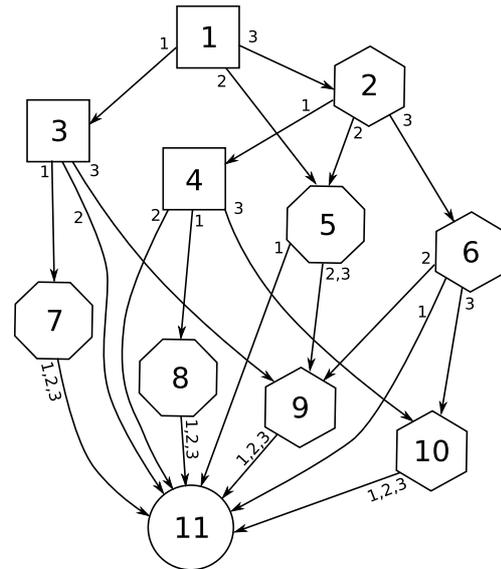
vstup

11
4 3 5 2
6 4 5 6
4 7 11 9
4 8 11 10
8 11 9 9
6 11 9 10
8 11 11 11
8 11 11 11
6 11 11 11
6 11 11 11

výstup

8

Mapa znázorňujúca vstup:



A-III-3 Prekladacie stroje

Študijný text nájdete za zadaním súťažnej úlohy. Je identický s tým z krajského a domáceho kola.

Súťažná úloha

- a) (6 bodov) Množina N obsahuje desiatkové zápisy kladných celých čísel.

Zostrojte prekladací stroj A s čo **najmenej stavmi**, pre ktorý platí: Nech X je ľubovoľná podmnožina N . Potom $A(X)$ je množina zápisov čísel, ktoré sú v X a zároveň nie sú deliteľné číslom 105.

Teda napríklad pre $X = \{47, 315, 411, 1050\}$ musí byť $A(X) = \{47, 411\}$.

Ak si myslíte, že takýto prekladací stroj neexistuje, dokážte to.

- b) (4 body) Množina X obsahuje jediný reťazec ε .

Množina Y obsahuje práve všetky reťazce, ktoré sú tvorené písmenami a a b a písmen a je v nich rovnako ako písmen b . Teda napríklad $\varepsilon \in Y$, $abba \in Y$, $aababb \in Y$, ale $a \notin Y$ a $bbab \notin Y$.

Zostrojte prekladací stroj s čo **najmenej stavmi**, ktorý preloží množinu X na Y .

Ak si myslíte, že takýto prekladací stroj neexistuje, dokážte to.



Študijný text

Prekladací stroj je stroj, ktorý dostáva ako vstup reťazec znakov. Tento reťazec postupne číta a podľa vopred zvolenej sady pravidiel (teda svojho programu) občas nejaké znaky zapíše na výstup. Po tom, ako stroj spracuje celý vstupný reťazec a úspešne skončí výpočet, zoberieme reťazec znakov zapísaný na výstup a nazveme ho **prekladom** vstupného reťazca.

Výpočet stroja nemusí byť jednoznačne určený. Inými slovami, sada pravidiel môže niekedy stroju umožniť rozhodnúť sa. V takomto prípade sa môže stať, že niektorý reťazec bude mať viac rôznych prekladov.

Naopak, môže sa stať, že v nejakej situácii sa podľa danej sady pravidiel nebude dať v preklade pokračovať vôbec. V takomto prípade sa môže stať, že niektorý reťazec nebude mať vôbec žiadny preklad.

Formálnejšia definícia prekladacieho stroja

Každý prekladací stroj pracuje nad nejakou vopred zvolenou konečnou množinou znakov. Túto budeme volať **abeceda** a značiť Σ . V súťažných úlohách bude vždy Σ známa zo zadania úlohy. Abeceda nikdy nebude obsahovať znak \$, ten budeme používať na označenie konca vstupného reťazca.

Stroj si môže počas prekladu reťazca pamätať konečne veľkú informáciu. Formálne toto definujeme tak, že stroj sa v každom okamihu prekladu nachádza v jednom z *konečne veľa* stavov. Nutnou súčasťou programu prekladacieho stroja je teda nejaká konečná **množina stavov**, v ktorých sa môže nachádzať. Túto množinu označíme K . Okrem samotnej množiny stavov je taktiež potrebné uviesť, v ktorom stave sa stroj nachádza na začiatku každého prekladu. Tento stav budeme volať **začiatočný stav**.

Program stroja sa bude skladať z konečného počtu prekladových pravidiel. Každé pravidlo je štvorica (p, u, v, q) , kde $p, q \in K$ sú nejaké dva stavy a u, v sú nejaké dva reťazce znakov z abecedy Σ .

Stavy p a q môžu byť aj rovnaké. Reťazec u môže ako svoj jediný znak obsahovať znak \$. Reťazce u a v môžu byť aj rovnaké a môžu byť aj prázdne. Aby sa program ľahšie čítal, budeme namiesto prázdneho reťazca písať ε .

Význam takéhoto pravidla je nasledujúci: „Ak je stroj práve v stave p a neprečítaná časť vstupu začína reťazcom u , môže tento reťazec zo vstupu prečítať, na výstup zapísať reťazec v a zmeniť stav na q .“ Všimnite si, že pravidlo tvaru (p, ε, v, q) môžeme použiť vždy, keď sa stroj nachádza v stave p , bez ohľadu na to, aké znaky ešte zostávajú na vstupe.

Ešte potrebujeme povedať, kedy preklad úspešne skončil. V prvom rade budeme vyžadovať, aby prekladací stroj dočítal celý vstupný reťazec. Okrem toho umožníme stroju „povedať“, či sa mu preklad podaril alebo nie. Toto spravíme tak, že niektoré stavy stroja nazveme **ukončovacie stavy**. Množinu ukončovacích stavov budeme značiť F .

Úplne formálna definícia prekladacieho stroja

Prekladací stroj je usporiadaná päťica (K, Σ, P, q_0, F) , kde Σ a K sú *konečné* množiny, $q_0 \in K$, $F \subseteq K$ a P je *konečná* množina prekladových pravidiel popísaných vyššie. Presnejšie, nech Σ^* je množina všetkých reťazcov tvorených znakmi zo Σ , potom P je *konečná* podmnožina množiny $K \times (\Sigma^* \cup \{\$\}) \times \Sigma^* \times K$. (Pre každé $q \in K$ budeme množinu pravidiel, ktorých prvá zložka je q , volať „prekladové pravidlá zo stavu q “.)

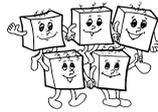
Ak teda chceme definovať konkrétny prekladací stroj, musíme uviesť všetkých päť vyššie uvedených objektov.

Keď už máme definovaný konkrétny stroj $A = (K, \Sigma, P, q_0, F)$, môžeme definovať, ako tento stroj prekladá konkrétny reťazec. Povieme najskôr formálnu definíciu, potom ju slovne vysvetlíme.

Množina **platných prekladov** reťazca u prekladacím strojom A je:

$$A(u) = \left\{ v \mid \begin{array}{l} \exists n \geq 0; \quad \exists (p_1, u_1, v_1, r_1), \dots, (p_n, u_n, v_n, r_n) \in P; \\ (\forall i \in \{1, \dots, n-1\}; r_i = p_{i+1}) \wedge p_1 = q_0 \wedge r_n \in F \wedge \\ \wedge \exists k \geq 0; u_1 u_2 \dots u_n = u \underbrace{\$ \dots \$}_k \wedge v_1 v_2 \dots v_n = v \end{array} \right\}$$

Definícia hovorí, kedy je reťazec v prekladom reťazca u . Vysvetlíme slovne význam jednotlivých riadkov definície:



- Prvý riadok hovorí, že aby sa dalo u preložiť na v , musí existovať nejaká postupnosť prekladových pravidiel, ktorú pri tomto preklade použijeme. Zvyšné dva riadky popisujú, ako táto postupnosť musí vyzeráť.
- Druhý riadok zabezpečuje, aby stavy v použitých pravidlách boli správne: Prvé pravidlo musí byť pravidlom zo začiatočného stavu, každé ďalšie pravidlo musí byť pravidlom z toho stavu, do ktorého sa stroj dostal použitím predchádzajúceho pravidla.
Navyše stav, v ktorom bude stroj na konci výpočtu, musí byť ukončovací.
- Posledný riadok hovorí o reťazcoch, ktoré stroj pri použití dotýčnych prekladových pravidiel prečíta a zapíše.
Reťazec, ktorý pri použití týchto pravidiel stroj prečíta zo vstupu, musí byť naozaj zadaný reťazec u , prípadne sprava doplnený vhodným počtom znakov $\$$.
Reťazec, ktorý stroj zapíše na výstup, musí byť presne reťazec v .

Na čo budeme používať prekladacie stroje?

Nám budú prekladacie stroje slúžiť na preklad jednej množiny reťazcov na inú množinu reťazcov. Ak A je prekladací stroj a $M \subseteq \Sigma^*$ nejaká množina reťazcov, tak preklad množiny M strojom A je množina $A(M) = \bigcup_{u \in M} A(u)$.

Inými slovami, výslednú množinu $A(M)$ zostrojíme tak, že zoberieme všetky reťazce z M a pre každé z nich dáme do $A(M)$ všetky jeho platné preklady.

Príklad 1

Majme abecedu $\Sigma = \{0, \dots, 9\}$. Nech M je množina všetkých tých reťazcov, ktoré predstavujú zápisy kladných celých čísel v desiatkovej sústave. Zostrojíme prekladací stroj A , pre ktorý bude platiť, že prekladom tejto množiny M bude množina zápisov všetkých kladných celých čísel, ktoré sú deliteľné tromi.

Riešenie

Najjednoduchšie bude jednoducho vybrať z M tie čísla, ktoré sú deliteľné tromi. Náš prekladací stroj bude kopírovať cifry zo vstupu na výstup, pričom si bude v stave pamätať, aký zvyšok po delení tromi dáva doteraz prečítané (a zapísané) číslo. Ak je po dočítaní vstupu v stave zodpovedajúcom zvyšku 0, prejde do ukončovacieho stavu.

Formálne A bude päťica $(K, \Sigma, P, 0, F)$, kde $K = \{0, 1, 2, \text{end}\}$, $F = \{\text{end}\}$ a prekladové pravidlá sú nasledovné:

$$P = \left\{ (x, y, y, z) \mid x \in \{0, 1, 2\} \wedge y \in \Sigma \wedge z = (10x + y) \bmod 3 \right\} \cup \left\{ (0, \$, \varepsilon, \text{end}) \right\}$$

Príklad 2

Majme abecedu $\Sigma = \{a, e, i, \bullet, \text{—}\}$. Zostrojíme prekladací stroj B , pre ktorý bude platiť, že prekladom ľubovoľnej množiny M , ktorá obsahuje len reťazce z písmen a, e a i , bude množina tých istých reťazcov v morzeovke (bez oddeľovačov medzi znakmi). Zápisy našich písmen v morzeovke sú nasledujúce: a je $\bullet\text{—}$, e je \bullet a i je $\bullet\bullet$.

Napríklad množinu $M = \{ae, eea, ia\}$ by náš stroj mal preložiť na množinu $\{\bullet\text{—}\bullet, \bullet\bullet\bullet\text{—}\}$. (Všimnite si, že reťazce eea a ia majú v morzeovke bez oddeľovačov rovnaký zápis.)

Riešenie

Prekladací stroj B bude jednoducho čítať vstupný reťazec po znakoch a zakaždým zapíše na výstup kód prečítaného znaku.

Formálne B bude päťica $(K, \Sigma, P, \heartsuit, F)$, kde $K = \{\heartsuit\}$, $F = \{\heartsuit\}$ a prekladové pravidlá vyzerajú nasledovne:

$$P = \{(\heartsuit, a, \bullet\text{—}, \heartsuit), (\heartsuit, e, \bullet, \heartsuit), (\heartsuit, i, \bullet\bullet, \heartsuit)\}$$

Všimnite si, že nepotrebujeme explicitne kontrolovať, či sme na konci vstupu. Totiž počas celého prekladu sme v ukončovacom stave, a teda akonáhle prečítame posledný znak zo vstupu, je zapísaný preklad platný.



Príklad 3

Majme abecedu $\Sigma = \{a, e, i, \bullet, \blacksquare\}$. Zostrojíme prekladací stroj C , pre ktorý bude platiť, že prekladom ľubovoľnej množiny M , ktorá obsahuje len reťazce zo znakov \bullet a \blacksquare , bude množina **všetkých** reťazcov z písmen a, e a i , ktoré keď zapíšeme v morzeovke (bez oddeľovačov medzi znakmi), dostaneme reťazec z M . Napr. množinu $M = \{\bullet\blacksquare\bullet, \bullet\bullet\bullet\blacksquare\}$ by náš stroj mal preložiť na $\{ae, eea, ia\}$.

Riešenie

Nášmu prekladaciemu stroju dáme možnosť sa v každom okamihu prekladu rozhodnúť, že ide čítať kód nejakého písmena a zapísať na výstup toto písmeno. Potom každej možnosti, ako rozdeliť vstupný reťazec na kódy písmen, bude zodpovedať jeden platný preklad.

Formálne C bude päťica $(K, \Sigma, P, \diamond, F)$, kde $K = \{\diamond\}$, $F = \{\diamond\}$ a prekladové pravidlá budú:
 $P = \{(\diamond, \bullet\blacksquare, a, \diamond), (\diamond, \bullet, e, \diamond), (\diamond, \bullet\bullet, i, \diamond)\}$.

Ako príklad si ukážeme, ako mohol prebiehať preklad reťazcov z vyššie uvedenej množiny M . Sú tieto tri možnosti:

$(\diamond, \bullet\blacksquare, a, \diamond), (\diamond, \bullet, e, \diamond)$
 $(\diamond, \bullet\bullet, i, \diamond), (\diamond, \bullet\blacksquare, a, \diamond)$
 $(\diamond, \bullet, e, \diamond), (\diamond, \bullet, e, \diamond), (\diamond, \bullet\blacksquare, a, \diamond)$

Ak by sme skúsili na ľubovoľnom vstupe z M používať prekladové pravidlá v inom poradí – napr. na vstupe $\bullet\bullet\bullet\blacksquare$ použiť trikrát pravidlo $(\diamond, \bullet, e, \diamond)$ – nepodariť sa nám dočítať vstupný reťazec do konca.

Príklad 4

Majme abecedu $\Sigma = \{a, b, c\}$. Nech množina X obsahuje práve všetky tie reťazce, v ktorých je rovnako a a b . Teda napr. $abbccac \in X$, ale $cbaa \notin X$.

Nech množina Y obsahuje práve všetky tie reťazce, ktoré neobsahujú žiadne a , neobsahujú podreťazec bc , a písmen c je dvakrát toľko ako písmen b . Teda napr. $ccccbb \in Y$, ale $cccbeb \notin Y$ a $acacba \notin Y$.

Zostrojíme prekladací stroj D , ktorý preloží X na Y .

Riešenie

Budeme prekladať len niektoré vhodné reťazce z množiny X , presnejšie to budú tie, v ktorých nie sú žiadne c a všetky a idú pred všetkými b . Takýto reťazec preložíme tak, že najskôr každé a prepíšeme na cc , a potom skopírujeme na výstup všetky b . Teda napr. prekladom slova $aabb$ bude slovo $ccccbb$.

Formálne D bude päťica $(K, \Sigma, P, \text{čítaj-}a, F)$, kde $K = \{\text{čítaj-}a, \text{čítaj-}b\}$, $F = \{\text{čítaj-}b\}$ a prekladové pravidlá vyzerajú nasledovne:

$P = \{(\text{čítaj-}a, a, cc, \text{čítaj-}a), (\text{čítaj-}a, b, b, \text{čítaj-}b), (\text{čítaj-}b, b, b, \text{čítaj-}b)\}$

Prečo tento prekladací stroj funguje? Ak vstupný reťazec obsahuje nejaké písmeno c , pri jeho prekladaní sa pri prvom výskyte c náš stroj zasekne. Preto takéto reťazce nemajú žiaden platný preklad. Podobne nemajú platný preklad reťazce, kde nie je dodržané poradie písmen a a b . Totiž akonáhle náš stroj prečíta prvé b , prejde do stavu $\text{čítaj-}b$, a ak sa teraz ešte na vstupe vyskytne a , stroj sa zasekne.

Platné preklady teda existujú naozaj len pre slová vyššie popísaného tvaru. Je zjavné, že z každého z nich vyrobíme nejaký reťazec z Y , preto $D(X) \subseteq Y$. A naopak, ak si povieme konkrétne slovo z Y , ľahko nájdeme slovo z X , ktoré sa naň preloží, preto $Y \subseteq D(X)$, a teda $Y = D(X)$.