1. príklad

Zostrojte viacpáskový deterministický Turingov stroj, ktorý na vstupnej páske dostane slovo predstavujúce binárny zápis čísla $x \in \{0,1\}^*$ a na jednu z pomocných pások zapíše a^n, kde $n = BinToDec(x)$. Môžete použiť ľubovoľné množstvo pomocných pások, nezabudnite ale označiť, kde sa nachádza výsledok. Všetky čítacie hlavy sa musia na konci výpočtu nachádzať za začiatku prislúchajúcich pások.

Poznámky k 1. príkladu

Vytvoríme si dvojpáskový TS. Vieme, že pozícia v binárnom číslo čítanom odzadu reprezentuje mocniny dvojky, počnúc nultou. Na prvú pomocnú pásku si preto budeme písať postupne mocniny dvojky, ktoré budú reprezentované prislúchajúcim počtom a-čok. Druhá pracovná páska bude slúžiť na zápis výsledku. Zostrojený TS si môžete stiahnuť a spustiť.

- Na vstupnej páske prečítame slovo do konca, tj. po dolár (stav q1).
- Na prvú pomocnú pásku (1.pp) si zapíšeme prvé a (reprezentujúce 2^0) a ak sme na vstupnej páske čítali 1, zapíšeme a aj na 2.pp. Ak tam bola 0, iba sa posunieme doľava (stavy q2 a q3).
- Kým existuje ďalšia pozícia v binárnom čísle na vstupe, budeme musieť vždy zvýšiť počet áčok na 1.pp na dvojnásobok, a v prípade, že čítame 1 pridať ho aj na 2.pp (na to slúžia stavy q4 až q8).
- Zdvojnásobovanie áčok budeme robiť od konca slova na 1.pp a to tak, že keď nájdeme a, prepíšeme ho na A a zapíšeme ešte jedno A na koniec pásky. Toto opakujeme kým neprídeme na cent (tj. nezduplikujeme všetky a) (stavy q5 a q6).
- Ak je na danej pozícii v binárnom čísle 1, na 1.pp sa hýbeme doprava a meníme A na a, a spolu s tým sa hýbeme aj na 2.pp a pridávame tam rovnaký počet a ako bol na 1.pp (stav q7).
- Ak je na danej pozícii v binárno čísle 0, iba prepíšeme všetky A na 1.pp na a a zostaneme s čítacou hlavou na poslednom a (stav q8).
- Ak sme na vstupnej páske prečítali celé slovo, vrátime všetky tri čítacie hlavy na začiatky pások a akceptujeme (stavy q9 a q10).

2. príklad

Zostrojte viacpáskový deterministický Turingov stroj, ktorý rozoznáva jazyk:

(a)* $L_1 = \{ww \mid w \in \{0,1\}^*\}$

(b)* $L_2 = \{ww^R \mid w \in \{0,1\}^*\}$

(c) $L_3 = \{a^{n^2} \mid n > 1, n \in N\}$

Poznámky k 2. príkladu

Keďže prvé dva príklady ste určite zvládli už minule pri obyčajných TS, teraz s možnosťou použiť pracovné pásky to už bude ľahké. V prvom príklade treba zistiť, kde je polovica slova a potom skontrolovať, či sú rovnaké. Napríklad ako v tomto riešení.

V béčku si môžete všimnúť, že každé slovo tvaru ww^R je palindrom, teda sa číta rovnako odpredu ako odzadu. Pozor, nie každý palindrom ale vieme napísať ako ww^R (napr. slovo 10001), musíte teda ešte overiť, že slovo má párnu dĺžku. Môžete ale použiť aj riešenie áčka a vhodne ho upraviť.

Idea riešenia 2(c): Ako by sme mohli rozoznávať slová, ktoré obsahujú počet áčok rovný druhej mocnine čísla väčšieho ako jedna? No vieme si mocniny predpočítať a potom iba skontrolovať, či na vstupe je slovo rovnakej dĺžky. A ako si ich predpočítame? Použijeme na to dve pomocné pásky – na prvej budeme mať n áčok, na druhej vždy vypočítame n^2. Vieme, že najmenšie slovo akceptované týmto TS je slovo $a^{2^2} = aaaa$, takže si ho na začiatku pripravíme na pásky.

Ako zmeníme a^{n^2} na $a^{(n+1)^2}$? Vypočítajme o koľko áčok viac bude mať dlhšie slovo. Tj. vypočítajme rozdiel $(n+1)^2 - n^2 = n^2 + 2n + 1 - n^2 = 2n + 1$. To znamená, že na 2pp iba dvakrát skopírujeme obsah 1pp, pridáme ešte jedno áčko a o jedno áčko zvýšime aj 1pp.

Vždy po vypočítaní druhej mocniny skontrolujeme, či nemáme rovnaký počet ako na vstupe. Ak je už na 2pp viac áčok ako na vstupe, slovo zamietneme (zjavne nie je druhou mocninou), ak je ich menej, spravíme druhú mocninu o jedna väčšieho čísla.

Vyskúšajte si moje riešenie. Viete zostrojiť aj iný postup? Skúste pre každý zistiť, koľkokrát budete musieť prejsť všetky pásky (tj. zistite zložitosť tohto algoritmu).

3. príklad

Zostrojte viacpáskový nedeterministický Turingov stroj, ktorý rozoznáva jazyk:

(a) $L_1 = \{a^{n2} \mid n \in N\}$

(b) $L_2 = \{ww^R \mid w \in \{0,1\}^*\}$

(c) $L_3 = \{ww \mid w \in \{0,1\}^*\}$;

Riešenie 3. príkladu

Riešenie časti (a): Deterministický TS pre takmer rovnaký[1] jazyk sme zostrojili v predchádzajúcom príklade. Akým spôsobom vieme použiť nedeterminizmus na rýchlejší výpočet? Nedeterminizmus pri Turingových strojoch, rovnako ako pri automatoch, znamená, že si vieme čosi „tipnúť". Predtým to bol prechod v automate, teraz sa vďaka tomu, že máme pásky, pridáva aj možnosť tipnúť si nejaké slovo na páske.

Zamyslime sa, čo by sme si mohli „tipovať". Mohlo by to byť rovno n^2, čiže iba skontrolujeme so vstupom a máme? Nie, toto fungovať nebude, lebo NTS akceptuje, ak aspoň jedna vetva výpočtu skončí v akceptačnom stave. Myslíte si, že napríklad pre slovo na vstupe aaa, by si NTS nemohol tipnúť aj aaa? Mohol, a tým pádom by sme ho akceptovali, aj keď by sme nemali. Treba si preto tipovanie predstaviť skôr ako funkciu random – vráti vám (v tomto prípade) nejaké celé číslo (v jednotkovej sústave), ktoré vieme použiť na to, aby sme vypočítali druhú mocninu.

Takže poďme vymyslieť algoritmus. Zostrojíme 2-páskový NTS. Na prvú pracovnú pásku si „tipneme" (nedeterministicky určíme) n, na druhú pracovnú pásku vypočítame n^2 (teraz zvolíme klasický postup, čiže $n^2 = n.n$) a skontrolujeme, či počet áčok na 2.pp je rovnaký, ako počet áčok na vstupe. Ak áno, akceptujeme. Nedeterminizmus nám zaručuje, že ak existuje také n, pre ktoré n^2 je rovné počtu áčok na vstupe, tak náš TS skončí v akceptačnom stave.

Môžete si stiahnuť riešenie v programe JFLAP. Jeho idea je nasledovná:

- Najprv si v stave q_1 "tipneme" počet áčok, ktoré zapíšeme na 1.pp.
- V stavoch q_2 až q_5 postupne počítame n^2 na 2.pp a to tak, že:
  - i) vezmem prvé a na 1.pp a prepíšem ho na A,

[1] v 2(c) je najkratšie akceptované slovo $aaaa$

ii) vrátim sa na začiatok 1.pp a na 2.pp píšem toľko áčok, koľko je znakov na 1.pp,

iii) keď ich prepíšem, nájdem ďalšie a a prepíšem ho na A, pokračujem bodom iii), pokiaľ všetky znaky na 1.pp nie sú A.

- Keď sme vypočítali n^2, kontrolujeme v stave q_6 vstup a 2.pp - na vstupnej páske sa hýbeme doprava a zároveň na 2.pp doľava. Ak na vstupnej páske narazíme na \$ a na 2.pp na ¢, tak sa na vstupe nachádza druhá mocnina a TS slovo akceptuje.

Všetky chýbajúce prechody by išli do stavu q_{reject}. Vieme, že NTS neakceptuje, ak ani jeden vstup neskončí v akceptačnom stave (tj. buď skončia v zamietajúcom stave, alebo budú všetky nekonečné). Takže aj bez nich NTS korektne rozoznáva jazyk. Skúste si spustiť v JFLAP zrýchlený výpočet, ak existuje nejaká akceptačná vetva výpočtu, zobrazí vám ju (dakedy ale potrebuje vygenerovať veľa vetiev).

Idea k 3(b): pri DTS sme museli zistiť, kde je stred vstupného slova a následne ho kontrolovať. V tomto prípade si to, kde je stred, môžeme tipnúť. Čiže – čítame vstupné slovo a píšeme ho na 1pp. Nedeterministicky sa rozhodneme, kedy sme už dočítali w a ideme čítať w^R. Vtedy prestaneme zapisovať na 1pp a zároveň s čítaním zvyšku vstupu čítame slovo na 1pp sprava doľava. Tým zároveň kontrolujeme, či sú rovnaké. Ak skončíme naraz na dolári (vstupná páska) a na cente (1pp), slová sú rovnaké a akceptujeme.

Idea k 3(c): Zamyslite sa, ako sa zmení riešenie z 3(b), skúste si oba TS navrhnúť a porovnať. Ktorý z nich prejde viackrát vstupnou a pracovnou páskou?

Poznámky k NTS

- Predstavme si, že TS je jaskyňa k pokladu (akceptačný stav), kde prechody sú chodby jaskyne. Chceme sa k pokladu dostať, ale niekde máme na výber viac križovatiek (prechodov), nedeterminizmus si tak môžete predstaviť dvoma spôsobmi – buď ako čarodejníka, ktorý má magickú guľu a na každej križovatke zistí kadiaľ ísť a tak nájde poklad. Alebo ako Alibabu, ktorý má mnoho zbojníkov a vždy keď je na výber viacero ciest, do každej pošle jedného. Ak sa aspoň jeden zbojník dostane k pokladu (akceptačnému stavu), tak ho zjavne získajú, ak ani jeden, tak nemajú nič.

- Povedané viac vysokoškolsky – NTS akceptuje vstup vtedy, keď *aspoň jedna* vetva výpočtu skončí v akceptačnom stave, a zamieta, keď *žiadna* *vetva* výpočtu neskončí v akceptačnom stave resp. *všetky vetvy* buď skončia v zamietajúcom stave alebo budú nekonečné.

- Na to je potrebné si dávať pozor pri navrhovaní vášho NTS, ak vám totiž aspoň jedna vetva výpočtu (jeden zbojník) skončí v akceptačnom stave, tak ste slovo na vstupe akceptovali, hoci ste to možno nechceli. Preto si radšej vždy odkrokujte aj nesprávne slová, či sa nijako nedostanete do akceptačného stavu.

- Pozor, nemôžete predpokladať, že magická guľa čarodejníka nájde konkrétny typ slov (napr. druhé mocniny, prvočísla, palindromy..), preto ju musíte skontrolovať. Tj. váš NTS si vie „tipnúť" napr. nejaký počet znakov, ale či ten počet vyhovuje vaším požiadavkám, si už musíte skontrolovať sami – tým zaručíte, že nebudete akceptovať nesprávny vstup.