1. príklad - TS počítajúce funkcie

Zostrojte Turingov stroj počítajúci funkciu, ktorá:

(a) na vstupe dostane slovo a^n a po výpočte na páske zostane a^{n+1}

(b) na vstupe dostane a^n a vráti b^n;

(c)* na vstupe dostane $a^x \# a^y$ a vráti a^{x+y};

(d) na vstupe dostane $a^x \# a^y$ a vráti a^{x-y}, pričom predpokladáme, že $x \geq y$;

(e)* na vstupe dostane a^n a vráti a^{2n};

Riešenie 1. príkladu

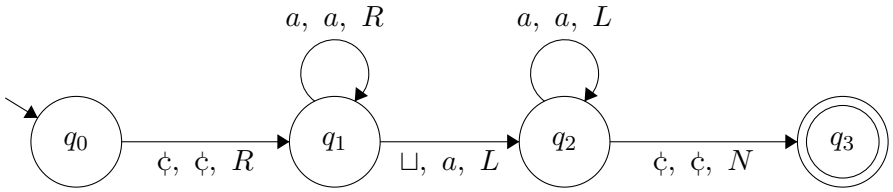
Ako začať pri konštruovaní TS? Musíme si najprv premyslieť stratégiu, čiže ako budeme postupovať, kedy sa čo stane... Je vhodné, ak si túto stratégiu napíšeme v bodoch. Následne sa treba zamyslieť nad tým, či sme pokryli všetky (dakedy aj okrajové) prípady, a ak nie, tak ich doplniť. Z definície TS vieme, že jeho prechody obsahujú výrazne viac vecí, ako konečné automaty – potrebujeme vedieť, čo na páske čítame, kam na nej posunieme čítaciu hlavu (doľava, doprava, alebo bude stáť) a čo na pásku (na to isté políčko) zapíšeme. Preto medzi každým stavom budeme musieť myslieť na všetky tri tieto akcie. Ukážme si to na príkladoch.

Príklad (a): zo zadania je jasné, že máme počet áčok na páske zvýšiť o jedna. Ako vyzerá páska na začiatku? Máme ¢, potom niekoľko a-čok a následne nekonečno veľa blankov ⊔. Náš postup by preto mohol byť taký, že prečítame všetky áčka a budeme sa hýbať doprava, až kým neprídeme na prvý blank. Ten prepíšeme na a a vrátime čítaciu hlavu na začiatok pásky (čo je dobrým zvykom pri konštruovaní takýchto TS).

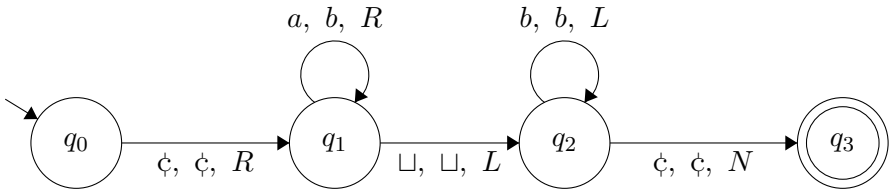
Takže, potrebujeme vstupný stav, tam prečítame cent a posunieme sa doprava na áčka[1]. Tam sa budeme „točiť" – čítať áčka, posúvať sa doprava, a áčka necháme na páske (t.j. ich „prepíšeme na a"). Keď prečítame blank, prepíšeme ho na a, hlava zostane stáť a posunieme sa do ďalšieho stavu. Tam čítaciu hlavu vrátime úplne naľavo (na začiatok pásky) a aby sme na páske nechali to, čo tam bolo, to čo čítame na ňu aj zapisujeme.

Výsledný TS je na obrázku, alebo si ho stiahnite.

[1] Pozor, cent pri TS NIKDY neprepisujeme, a ideálne, ani nič neprepisujeme na cent, keďže tým by sme akoby „odstihli" pásku.



Príklad (b): v tomto príklade máme áčka prepísať na béčka. To je pomerne jednoduché. Vždy, keď prečítame áčko, prepíšeme ho na béčko a posunieme sa doprava. Toto opakujeme, kým neprečítame blank. Následne sa bez prepisovania vrátime na začiatok pásky. Výsledný TS je na obrázku, alebo si ho stiahnite. Ako by sme museli zmeniť TS, ak by na vstupe bolo slovo nad abecedou $\{a, b\}$ a chceli by sme prepísať áčka na béčka? A čo by sme chceli zameniť áčka za béčka a béčka za áčka?



Príklad (c): nechávam na vás. Zamyslite sa, čo treba spraviť s mriežkou, aby ste dostali výsledok.

Príklad (d): kým v predchádzajúcom príklade sa stačilo iba jednoducho zbaviť mriežky, pri odčítavaní budeme musieť po páske prejsť viackrát. Za každé áčko napravo od mriežky budeme musieť jedno áčko naľavo „vymazať". Ako to spravíme? Možností je viacero. Popíšem jednu: nájdeme mriežku, doprava za ňou je áčko. To vymažeme, ale aby sme vedeli, že za ním ešte môžu byť áčka, neprepíšeme ho na blank, ale na ďalšiu mriežku. Teraz sa budeme vracať doľava, kým nenarazíme na prvé áčko pred mriežkou. Toto opäť prepíšeme na mriežku – čím ho vymažeme, ale zároveň budeme vedieť, že pred ním ešte môžu byť nejaké áčka. Páska bude obsahovať slovo v tvare $a^i\#^j a^k$, čiže nejaké áčka, nejaké mriežky a nejaké zvyšné áčka, ktoré ešte musíme odčítavať. Aby sme mali lepšiu predstavu, čo sa deje, skúsme si odkrokovať, ako bude vyzerať naša páska na tomto príklade:

$$¢aaaaa\#aaa␣ \tag{1}$$

$$¢aaaaa\#\#aa␣ \tag{2}$$

$$¢aaaa\#\#\#aa␣ \tag{3}$$

$$¢aaaa####a␣ \tag{4}$$

$$¢aaa#####a␣ \tag{5}$$

$$¢aaa######␣ \tag{6}$$

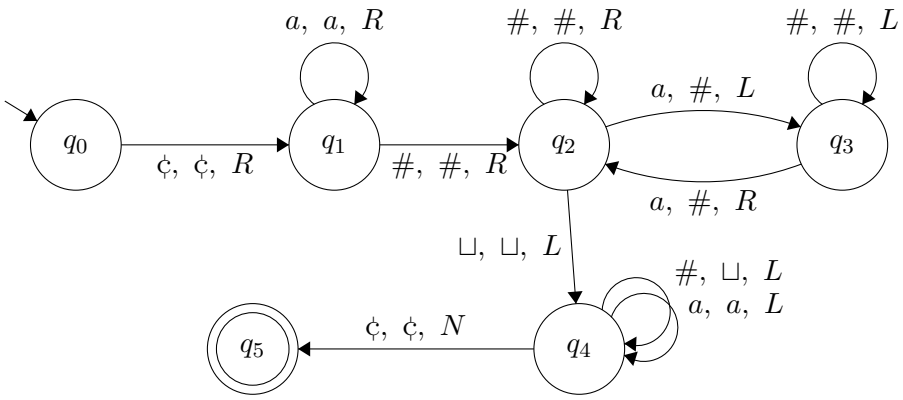
$$¢aa#######␣ \tag{7}$$

Ako budeme vedieť, že už sme všetky áčka za mriežkou odčítali? Keď budeme hľadať ďalšie a namiesto neho nájdeme blank, pretože to znamená, že už sme na konci nášho slova na páske. Počet áčok je už v poriadku, ale mali by sme sa zbaviť mriežok. Keďže aktuálne sme na prvom blanku, cestou doľava všetky mriežky prepíšeme na blanky, áčka necháme áčkami a celý výpočet skončíme, keď sa dostaneme na cent.

Čo myslíte, mohli by sme áčka prepisovať na blanky? Ak áno, prečo a ak nie, prečo? Vyskúšajte si upraviť tento TS, aby to robil. Treba ešte nejaké ďalšie úpravy? Viete potom jasne určiť, kedy už sme dokončili výpočet?

Viete vymyslieť aj iný spôsob, ako túto úlohu riešiť? Skúste pre oba spôsoby zistiť ich náročnosť – koľkokrát sa budete musieť presúvať po páske? Budú tieto prechody rovnako dlhé v oboch spôsoboch? Vyskúšajte si pre tento spôsob navrhnúť TS a odkrokujte ho na rovnakých slovách, aké skúsite na našom príklade.

Riešenie na obrázku alebo na stiahnite.



Príklad 1(e): na vstupe dostaneme n áčok a máme vrátiť dvakrát viac. Pre každé áčko na vstupe, tak potrebujeme dopísať na pásku ešte jedno. Znie to jednoducho, nájdeme áčko, nejako zaznačíme, že sme ho zarátali, ideme na koniec slova, pridáme tam a, vrátime sa a toto opakujeme. Ale ako zaručíme, že budeme dopisovať áčka iba za tie, ktoré tam boli už predtým a nie za tie, ktoré sme dopísali? Možno tak, že ich nebudeme priamo písať ako a, ale napríklad ako A.

Čiže, postup je nasledovný: prečítame prvé áčko za centom, prepíšeme ho na X. Prejdeme na prvý blank a napíšeme tam A. Vrátime sa po malých áčkach doľava, kým nenarazíme na X, z neho sa posunieme doľava na a, ktoré prepíšeme na X a posúvame sa po blank. Pozor, teraz bude na páske $¢X^i a^j A^k \sqcup$, čiže ak sa chceme dostať na blank, budeme musieť prečítať aj nejaké veľké Áčka. Rovnaké platí aj pre spätné hľadanie X. Toto budeme opakovať dokedy? Pokiaľ sa z X neposunieme doprava a tam namiesto a bude A. Čo znamená, že sme už zdvojnásobili každé pôvodné áčko. Teraz potrebujeme iba všetky X a A na páske prepísať na a. Nájdeme preto koniec slova (prvý blank), a odtiaľ pôjdeme doľava a budeme X aj A prepisovať na a.

Skúste si tento TS skonštruovať a porovnajte s tým mojím.

Myslíte si, že by postup fungoval aj keby sme pôvodné áčka neprepisovali na X ale na A? Viete vymyslieť aj iný spôsob, ako toto spraviť? Skúste porovnať tieto prístupy. Ktorý bude efektívnejší?

2. príklad - Obyčajné TS

Zostrojte Turingov stroj, ktorý vracia TRUE/FALSE (teda apkceptuje alebo zamieta), podľa toho, či slovo na páske patrí do jazyka:

(a) $L_1 = \{|w|_0 \ mod \ 2 = 0 \mid w \in \{0,1\}^*\}$

(b) $L_2 = \{w\#w \mid w \in \{a\}^*\}$

(c) $L_3 = \{w\#w \mid w \in \{a,b\}^*\}$

(d)* $L_4 = \{ww^R \mid w \in \{a,b\}^*\}$

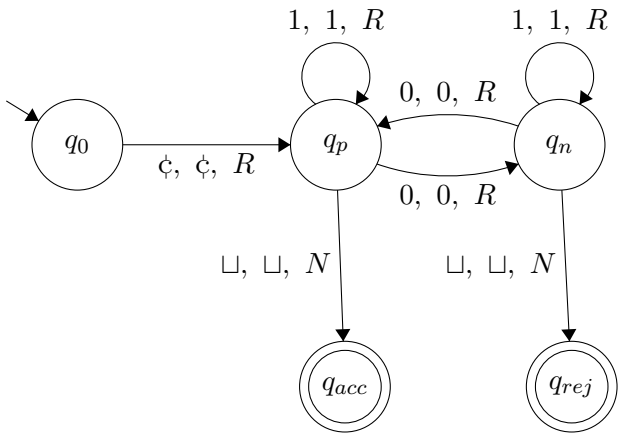
(e)* $L_5 = \{ww \mid w \in \{a,b\}^*\}$

(f)* $L_6 = \{|w|_a = |w|_b \mid w \in \{a,b\}^*\}$;

Riešenie 2. príkladu

Riešenie 2(a): jazyk L_1 je určite regulárny, čo znamená, že vieme vytvoriť DKA, ktorý ho bude rozoznávať. Ten vieme veľmi ľahko prerobiť na TS, v ktorom nepotrebujeme nič zapisovať. Zmena oproti DKA je tá, že na konci slova máme blank. A ten môže byť na konci slova z jazyka – vtedy slovo akceptujeme, takže pôjde do stavu q_{accept}, a ak slovo z jazyka nebude, skončí v stave q_{reject}.

Z obrázku vidíme, že sme museli pridať iba tieto dva stavy a vstup, kde sme prečítali cent. Vyskúšajte si riešenie.



Čo myslíte, vieme každý DKA prerobiť na TS? Vieme zostrojiť TS pre každý regulárny jazyk?

Riešenie 2(b): v tomto príklade už vidíme jazyk, ktorý nie je regulárny. Nevieme teda zostrojiť automat, ktorý by ho rozpoznával. Ukážme si, že pomocou TS to už nebude problém.

Chceme zistiť, či počet áčok pred mriežkou je rovnaký ako počet áčok za mriežkou. (Opäť, existuje veľa možných riešení, skúste potom vymyslieť nejaké iné.) Môj postup bude nasledovný – nájdem áčko tesne pred mriežkou, prepíšem ho na mriežku, a budem hľadať prvé áčko za mriežkou, ktoré tiež prepíšem na mriežku. Vrátim sa pred mriežku a pôjdem znovu. Vždy tak budem mať slovo v tvare áčka-mriežky-áčka. Toto budem opakovať dovtedy, kým sa mi áčka na jednej alebo druhej strane neminú. Keď sa minú na oboch stranách (čiže moje slovo sa bude skladať iba z mriežok), slovo akceptujem, ak na nejakej strane ešte áčko zostane (tj. buď $a^i\#^j$ alebo $\#^k a^m$), slovo zamietnem.

Vyskúšajte spustiť na riešení rôzne slová. Čo myslíte, všetky vetvy sú potrebné? Existujú tam stavy, do ktorých sa nikdy nedostanete? Odstráňte ich.

Riešenie 2(c): vedeli by sme upraviť TS z 2(b) tak, aby mohli byť slová nad abecedou $\{a, b\}$? Asi nebudeme môcť použiť rovnaký prístup, pretože napríklad pre slovo $ab\#ab$ by sme porovnávali a za mriežkou s b pred mriežkou, čo nie je dobrý prístup. Budeme preto musieť porovnávať písmená za centom s písmenami za mriežkou. Už sme spomínali, že v pekných riešeniach nič neprepisujeme na cent, preto písmená zo začiatku pásky budeme prepisovať napríklad na hviezdičku *. Budeme potrebovať aj dve vetvy – ak sme našli za centom (neskôr za hviezdičkami) a, a ak sme našli b. Podľa toho

budeme vedieť, aký znak hľadať za mriežkou (mriežkami). Zhrňme si to:

- nájdeme prvý znak za centom (neskôr hviezdičkou):
  - ak je to a alebo b, prepíšeme ho na hviezdičku a posúvame sa doprava až za mriežku:
    - ak tam bude rovnaký znak, prepíšeme ho na mriežku a vraciame sa doľava, kým nenarazíme na hviezdičku, a opakujeme celý postup.
    - ak tam bude iný znak alebo blank, môžeme rovno slovo zamietnuť.
  - ak je ďalší znak mriežka, znamená to, že na začiatku už žiadne písmená nemáme. Skontrolujeme, či na konci sú už iba mriežky:
    - ak nájdeme a alebo b zamietame (za mriežkou sa nachádza dlhšie slovo ako pred ňou),
    - ak nájdeme iba blanky, zjavne sú slová rovnaké a akceptujeme.

Vyskúšajme postup na pár slovách:

1) $¢abab\#abab\sqcup \longrightarrow ¢abab\#abab\sqcup \longrightarrow ¢*bab\#abab\sqcup \longrightarrow ¢*bab\#\#bab\sqcup \longrightarrow ¢**ab\#\#bab\sqcup \longrightarrow ¢**ab\#\#\#ab\sqcup \longrightarrow ¢***b\#\#\#ab\sqcup \longrightarrow ¢***b\#\#\#\#b\sqcup \longrightarrow ¢****\#\#\#\#b\sqcup \longrightarrow ¢****\#\#\#\#\sqcup \longrightarrow$ ACCEPT

2) $¢abab\#aba\sqcup \longrightarrow ¢abab\#aba\sqcup \longrightarrow ¢*bab\#aba\sqcup \longrightarrow ¢*bab\#\#ba\sqcup \longrightarrow ¢**ab\#\#ba\sqcup \longrightarrow ¢**ab\#\#\#a\sqcup \longrightarrow ¢***b\#\#\#a\sqcup \longrightarrow ¢***b\#\#\#\#\sqcup \longrightarrow ¢****\#\#\#\#\sqcup \longrightarrow$ REJECT

3) $¢aba\#abab\sqcup \longrightarrow ¢aba\#abab\sqcup \longrightarrow ¢*ba\#abab\sqcup \longrightarrow ¢*ba\#\#bab\sqcup \longrightarrow ¢**a\#\#bab\sqcup \longrightarrow ¢**a\#\#\#ab\sqcup \longrightarrow ¢***\#\#\#ab\sqcup \longrightarrow ¢***\#\#\#\#b\sqcup \longrightarrow$ REJECT

3) $¢aba\#abb\sqcup \longrightarrow ¢aba\#abb\sqcup \longrightarrow ¢*ba\#abb\sqcup \longrightarrow ¢*ba\#\#bb\sqcup \longrightarrow ¢**a\#\#bb\sqcup \longrightarrow ¢**a\#\#\#b\sqcup \longrightarrow ¢***\#\#\#b\sqcup \longrightarrow$ REJECT

V prvom slove skončíme, keď po hviezdičke vpravo bude mriežka. Prejdeme zvyšok pásky a keďže nenájdeme žiaden iný znak ako mriežku, slovo akceptujeme.

V druhom slove budeme hľadať béčko za mriežkami, ale nájdeme tam iba blank, čiže slovo pred mriežkou je dlhšie, zamietame.

V treťom slove skončíme, keď po hviezdičke vpravo bude mriežka. Skontrolujeme zvyšok pásky a nájdeme na konci b. Slovo za mriežkou je dlhšie ako slovo pred mriežkou, zamietame.

V štvrtom slove budeme hľadať za mriežkou a, keďže to sme prepísali pred mriežkou posledné, nájdeme tam ale b. Slová nie sú rovnaké, preto zamietame.

Vyskúšajte si podľa tohto postupu zostrojiť TS, alebo si ho stiahnite a skontrolujte.

3. príklad - Viacpáskové TS

Zostrojte viacpáskový Turingov stroj, ktorý rozoznáva jazyk:

(a) $L_1 = \{w \in \{0,1\}^* \mid |w|_0 = |w|_1\}$

(b) $L_2 = \{w \in \{0,1\}^* \mid |w|_0 = 3|w|_1\}$

(c)* $L_3 = \{w \in \{0,1,2\}^* \mid |w|_1 = 2|w|_2 = |w|_3\}$;

(d)* $L_4 = \{ww \mid w \in \{0,1\}^*\}$

(e)* $L_5 = \{ww^R \mid w \in \{0,1\}^*\}$

Riešenie 3. príkladu

Skôr ako začneme riešiť, zopakujeme si jednotlivé pojmy. Obyčajné Turingove stroje sú stroje s jednou vstupno-výstupnou páskou, ktorá začína centom a za vstupným slovom je nekonečne veľa blankov. Viacpáskové Turingove stroje obsahujú jednu vstupnú pásku, ktorá začína centom a končí dolárom. Táto vstupná páska je read-only, to znamená že na ňu nevieme nič zapisovať. Ďalej máme viac pomocných (pracovných) pások, ktoré obsahujú na začiatku iba cent a nekonečne veľa blankov. Na tieto môžeme písať, mazať, môžeme tam zapisovať výsledok a podobne. Jednopáskový TS[2] bude stroj, ktorý má jednu vstupnú a jednu pomocnú (pracovnú) pásku. Dvojpáskový bude mať dve pomocné (pracovné) pásky. Každá páska má vlastnú čítaciu hlavu, čo znamená, že ak sa na jednej hýbeme, na ďalšej môžeme aj stáť alebo sa hýbať opačným smerom (samozrejme, v rámci pravidiel, že nejdeme pred cent a za dolár). Z toho vyplýva, že každý prechod bude v

[2] v JFLAP treba vybrať dvojpáskový – do počtu pások rátajú aj vstupnú

tvare[3] *čítam, posúvam | čítam, píšem, posúvam | čítam, píšem, posúvam ...* - keďže na vstupnej páske nevieme nič prepisovať.

Riešenie (a): chceme zistiť, či vo vstupnom slove je rovnaký počet núl ako jednotiek. Pri obyčajnom TS sme museli prechádzať pásku veľakrát (ako napr. v 1 (d)), ale keďže máme pomocné pásky, asi ich vieme využiť na to, aby sme to zistili už počas jedného prechodu vstupným slovom. Mohli by sme napríklad pre každú nulu na vstupe zapísať jednu nulu na prvú pomocnú pásku (1pp) a za každú jednotku jednu nulu z 1pp vymazať. Bolo to to dobré riešenie?

Pozrime sa, čo by sa stalo so slovom 010101 – najprv by sme na 1pp napísali 0, potom ju vymazali, potom opäť napísali 0, zas vymazali a ešte raz napísali nulu a vzápätí ju vymazali. 1pp by tak zostala prázdna (až na cent) a slovo by sme mohli akceptovať. Čo by sa ale stalo pri slove 101010? Je to tiež slovo, ktoré by malo byť akceptované, ale teraz by sme sa najprv snažili mazať niečo z prvej pásky, hoci by ešte bola prázdna. Čo nie je dobrý prístup.

Dobre, možno ste si povedali, že to, či zapisujeme na 1pp nuly alebo jednotky bude závisieť od toho, čo je prvé na vstupe. Tak si zas ukážme slovo, ktoré nám tento nápad pokazí. Napríklad 100011 – na 1pp by sme písali jednotky, ale hneď na treťom znaku by sme sa snažili vymazať znak z prázdnej pásky. Takže asi ani toto nie je korektný postup.

Možno to len nevieme spraviť pri jednom prechode úplne jednoducho. Tak skúsme spraviť iné riešenie, ktoré možno nebude tak efektívne, a následne sa skúsime vrátiť k tomu, ako to spraviť jedným prechodom.

Čo keby sme si na 1pp napísali nuly zo vstupného slova, pri jednotkách na vstupe by sme sa iba posunuli doprava. Keď by sme prišli na koniec vstupného slova, vracali by sme sa po vstupnej páske doľava a za každú jednotku by sme vymazali 0 z 1pp. Bol by tento postup dobrý pre všetky tri prechádzajúce slová? Áno. Pri všetkých z nich by sme najprv napísali tri nuly na 1pp a následne, pri prechode páskou sprava doľava, by sme ich všetky tri vymazali. Tak by nakonci zostala 1pp prázdna. Čiže náš TS by akceptoval slovo vtedy, keď by sme sa na vstupnej páske dostali na cent a zároveň by bola 1pp prádzna (teda by sme sa tam tiež dostali na cent).

Kedy by náš TS zamietal? Ak by sme chceli mazať niečo z prázdnej 1pp.

[3] v JFLAP zas nastáva trochu zmena, keďže v ich TS môžete na vstupnú pásku písať, treba tam dávať aj tieto prechody, tj. x, x, SMER, kde x je nejaký znak. Aby ste vedeli vytvoriť TS podľa našej definície, dajte si pozor, aby ste na vstupnej páske nič neprepísali, a pri testovaní zadávajte aj okraje pásky ¢xxx\$, resp. na pomocné pásky dajte vždy na začiatok ¢. Ja vo svojich TS ¢ nahrádzam c a \$ S.

Alebo ak by na 1pp ešte zostali nuly, ale na vstupe by sme sa už dostali na cent. Skúsme si celý postup spísať v bodoch:

- čítame vstupnú pásku zľava doprava
  - ak čítam na vstupe 0, posuniem sa doprava, na 1pp napíšeme 0 a posunieme sa doprava,
  - ak čítam na vstupe 1, posuniem sa doprava, na 1pp nepíšeme nič a stojíme,
- keď prídeme na dolár na vstupe začneme čítať vstupnú pásku sprava doľava
  - ak čítame na vstupe 0, posuniem sa doľava, na 1pp nerobíme nič a stojíme (stáť môžeme na 0, ale aj na cente – napr. pre slovo 0011),
  - ak čítame na vstupe 1, posuniem sa doľava, na 1pp:
    - ak je tam 0, vymažeme ju a posunieme sa doľava,
    - ak je tam už cent, slovo zamietame (v slove bolo viac jednotiek ako núl)
  - ak čítame na vstupe cent, stojíme a na 1pp:
    - ak je tam tiež cent, slovo akceptujeme (má rovnaký počet núl a jednotiek),
    - ak je tam nula, slovo zamietame (v slovo bolo viac núl ako jednotiek)

Teraz nám už len zostáva tento TS zostrojiť. Skúste ísť podľa návodu a následne porovnajte s tým mojím. Nezabudnite definovať celý TS – tj. aj vstupnú abecedu $\{0, 1\}$ (z čoho sa skladá vstupné slovo) a pracovnú abecedu $\{¢, \$, \sqcup, 0, 1, \sqcup\}$ (aké všetky znaky vieme prečítať). Zamyslite sa, čo sa stane s prázdnym slovom na vstupe. Bude akceptované a má byť akceptované?

Napadli vám aj iné možnosti ako tento príklad riešiť? Určite sa to dá pomocou dvojpáskového TS, kde na jednej páske budete mať nuly a na druhej jednotky a potom iba skontrolujete, či ich je rovnako veľa. Prípadne iné obmeny. No ale dá sa to vyriešiť aj tak, že po vstupnom slove prejdeme naozaj iba raz a budeme mať iba jednu pomocnú pásku? Áno, dá. Na začiatku sme zistili, že ak by sme si zapisovali na 1pp nuly a za každú jednotku by sme jednu nulu z 1pp vymazali, problematické by bolo, keby sme mali prázdnu pásku a chceli by sme mazať. To môžeme vyriešiť napríklad tak, že keď je

páska prázdna, ale mali by sme jeden symbol vymazať, napíšeme si na pásku napríklad −. Následne, ak ideme zapísať nulu ale už je tam mínus, toto mínus vymažeme. Tu si však treba dávať pozor na to, ako sa hýbe hlava na 1pp, kedy mažeme, kedy sa kam posúvame a kedy určite vieme, že môžeme akceptovať a kedy zamietať. Skúste si taký TS navrhnúť, prípadne sa môžete inšpirovať tým mojím.

Riešenie (b): ako budeme kontrolovať slová, v ktorých je trikrát viac núl ako jednotiek? Môžeme upraviť nejaký z postupov v predchádzajúcom príklade? Samozrejme. Tak skúsme najprv načrtnúť, ako by sme tento príklad riešili s jednopáskovým TS.

Najprv vstupné slovo čítame zľava doprava. Za každú prečítanú jednotku na 1pp zapíšeme tri jednotky. Pri prečítaní nuly sa iba posunieme doprava a na 1pp nerobíme nič. Keď narazíme na dolár, začneme slovo čítať sprava doľava. Teraz vždy, keď prečítame nulu, jednu jednotku z 1pp vymažeme (tiež vymazávame sprava doľava). Keďže sme si na začiatku strojnásobili počet jednotiek, slovo akceptujeme, ak sa na oboch páskach dostaneme naraz na cent. Ak bude 1pp prázdna skôr ako sme stihli prečítať celé vstupné slovo, slovo má viac núl ako tri krát počet jednotiek. Ak sa na vstupnej páske dostaneme na cent a na 1pp budú ešte jednotky, tak slovo bude mať menej núl ako tri krát počet jednotiek. V oboch týchto prípadoch zamietame.

Ako by vyzeral dvojpáskový TS? Na jednu pásku by sme si mohli napísať nuly, na druhú jednotky. A tu máte zas dve možnosti – buď každú jednotku zapíšete trikrát a následne iba skontrolujete, či sú slová na 1pp a 2pp rovnako dlhé. Alebo zapíšete iba reálne počty núl a jednotiek a pri následnej kontrole za každú jednotku na 2pp prejdete tri nuly na 1pp. Vyskúšajte si jeden z týchto spôsobov zostrojiť.

Riešenie (c): prepokladám, že už asi rovno viete, ako by ste riešili tento príklad. Najjednoduchšie by, samozrejme, bolo použiť tri pomocné pásky. Na jednej mať nuly, na druhej jednotky a na tretej dvojky. Ale vieme to vyriešiť aj s dvoma pracovnými páskami? Áno, ak trochu skombinujeme oba prístupy z predchádzajúcej úlohy.

Na 1pp si budeme písať nuly – za každú nulu na vstupe sem napíšeme jednu nulu. Na 2pp za každú jednotku na vstupe napíšeme dve jednotky. No a následne, keď už budeme na konci slova, budeme odzadu kontrolovať – ak prečítame dvojku, vymažeme jednu nulu z 1pp a jednu jednotku z 2pp. Ak prečítame na vstupe všetky dvojky a obe pracovné pásky budú prázdne, slovo akceptujeme. Vyskúšajte si tento TS skonštruovať.

Poznámky k (d) a (e) V tomto 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.

4. príklad - Ďalšie TS

(a) Zostrojte dvojpáskový Turingov stroj, ktorý bude simulovať Stack (zásobník).

vstupnej páske dostane $\uparrow$, a, b, kde $\uparrow$ znamená POP, a symboly a, b PUSH daných symbolov do zásobníka,

- na prvej pracovnej páske bude aktuálny stav zásobníka,
- na druhej pracovnej páske výstup z funkcie POP,
- TS skončí v akceptačnom stave, ak po vykonaní celého vstupu zostane zásobník prázdny.
- *Je na vás, čo spraví TS, ak je prázdny zásobník a zavolá sa POP. Môže to „vyhodiť error", teda presunúť sa do Reject-u, alebo iba vrátiť blank.*

(b)* Na vstupe dostanete postupnosť 0 a 1, ktoré predstavujú jamy (0) a kopy (1) na jednom úseku. Ak sa jama a kopa nachádzajú vedľa seba, vieme jamu zasypať hlinou z kopy a obe zaniknú. Vašou úlohou je zistiť, koľko nerovností (jám alebo kôp) zostane na konci na úseku, ak vykonáme všetky operácie zasyp, ktoré sú možné. Napr. ak máme 01000111 nezostane na konci žiadna nerovnosť, pri 0100 zostanú dve, a pri 11100 jedna. Na výstupe vráťte a^n, kde n je počet nerovností. Navrhnite DTS, ktorý bude riešiť túto úlohu[4]

(c) Vytvorte DTS, ktorý na vstupe dostane slovo tvaru $c_1z_1c_2z_2...z_{m-1}c_m$, kde $c_i \in \{a\}^+$ a $z_i \in \{+,-\}$ pre $1 \leq i \leq m$. Slovo interpretujeme ako výraz obsahujúci čísla $|c_i|_a$ oddelené znamienkami plus a mínus. TS slovo akceptuje, ak interpretácia slova je rovná 0, inak zamieta.

Idea riešenia 4.a

Príklad sa dá vyriešiť presne podľa zadania. Vždy, keď bude na vstupe symbol a alebo b, napíšeme ich na 1pp a posunieme sa na nej doprava. Ak

[4] úloha je z KSP, nájdete ju na `https://www.ksp.sk/ulohy/zadania/1420/`

tam bude ↑, nájdeme na 1pp posledný symbol, napíšeme ho na 2pp a z 1pp ho vymažeme. Keď prejdeme celé slovo na vstupnej páske, skontrolujeme, či je 1pp (zásobník) prázdna. Vyskúšajte, prípadne si pozrite moje riešenie.

Vedeli by sme vytvoriť TS, ktorý by akceptoval také slová nad abecedou $\{\uparrow, a, b\}$, že ich vykonaním zostane prázdny zásobník, aj bez toho, že by sme si zapisovali aktuálny stav zásobníka? Mohli by ste navrhnúť, že veď iba stačí zistiť, či je počet znakov a, b rovnaký ako počet ↑ v slove. To by ale nebola pravda napríklad pre slovo $\uparrow\uparrow ab$, pri ktorom by na konci zostalo v zásobníku ab, alebo by už rovno vyhodil error pri pokuse o POP z prázdneho zásobníka. V tomto prípade je dôležité poradie operácií, preto vhodné riešenie je práve simulovanie zásobníka (samozrejme, ak by sme iba zisťovali, či je slovo z jazyka, nepotrebovali by sme 2pp).

Idea riešenia 4.c

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).

Poznámky

Obyčajné Turingove stroje

- Pri konštrukcii Turingovho stroja si najprv rozmyslite celú ideu toho, ako bude pracovať a načrtnite ju. Postupne riešte „dobré" slová, vysvetlite, ako bude páska vyzerať po aplikovaní navrhnutých akcií, ako na to bude váš TS reagovať, aké máte krajné prípady a kde ich ošetrujete, ako zistíte, že je slovo z jazyka (t.j. kedy sa dostane do akceptačného stavu)... Zamyslite sa, či ste vyriešili naozaj všetky možnosti. Ak niečo predpokladáte, napíšte to (napr. *na vstupe je korektné slovo* a pod.).

- TS je vlastne algoritmus, alebo program. Popíšte ho preto tak, aby si čitateľ vedel zostrojiť vlastný TS iba podľa vášho opisu.

- Ak medzi dvoma stavmi (alebo na jednej slučke) viac prechodov rovnakého typu, napr. a, a, R, b, b, R, c, c, R, môžete ich napísať ako jeden použitím substitúcie. Na prechod napíšete x, x, R a k nemu $x \in \{a, b, c\}$. Pre prechody $0, \sqcup, L$ a $1, \sqcup, L$ sa dá zas použiť substitúcia $y, \sqcup, L$, kde $y \in \{0, 1\}$ atď.

- Vhodne si pomenujte stavy, označte si časti TS, ak to potrebujete, a popisujte ich zvlášť. Ak používate substitúciu, buďte jednotní v celom stroji a napíšte ju niekam na viditeľné miesto (sprehľadní to váš diagram).

- TS v každom stave niečo z pásky číta, niečo na ňu zapisuje a hýbe sa vpravo, vľavo alebo stojí. Determinizmus pri TS určuje to, že v prípade, že v stave q prečítame symbol e, vieme jednoznačne určiť, aký symbol zapíšeme a kam sa pohneme. Zjavne teda nevieme mať z jedného stavu zároveň takéto prechody e, e, R a e, e, L, či e, a, R.

- Ak sa raz do akceptačného stavu dostaneme, TS slovo akceptuje bez ohľadu na to, či sme ho prečítali celé alebo nie (t.j. z akceptačného stavu sa nevieme dostať preč). To isté platí aj pre zamietací stav.

- Cent nemažeme, neprepisujeme na nič iné a nič iné neprepisujeme na cent. Páska je nekonečná iba jedným smerom (doprava), ľavý koniec

pásky je označený centom. Za vstupným slovom nasleduje nekonečno blankov.

Viacpáskové Turingove stroje

Hlavný rozdiel medzi obyčajným TS a viacpáskovým TS je v tom, že pri viacpáskovom je vstupná páska READ-ONLY - začína ¢ a končí $. Pracovné pásky číslujeme od 1 a na začiatku sú prázdne (až na ¢) a čítacie hlavy na začiatku. Pre každý krok viacpáskového TS musíme definovať:

- čo čítame na vstupnej páske a kam sa posúvame;
- pre každú pracovnú: čo čítame, čo zapisujeme a kam sa posúvame.

Determinizmus pri viacpáskových TS značí, že ak v jednom stave prečítame určitú sadu symbolov na páskach, jednoznačne bude stroj vedieť, kam sa posunúť. Teda nie je povolené mať zároveň prechody ako napr.: $a, R \mid b, b, R \mid c, c, L$ a prechod $a, N \mid b, b, R \mid c, c, L$