Cvičenie 8 - Turingove stroje

Ing. Viliam Hromada, PhD.

C-510
Ústav informatiky a matematiky
FEI STU

viliam.hromada@stuba.sk

25.11.2020

Príklad č. 1

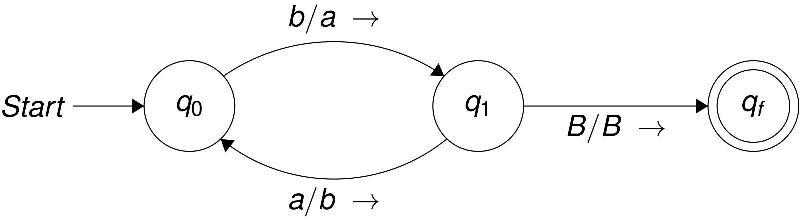
Je daný Turingov stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, kde

- Množina stavov $Q = \{q_0, q_1, q_f\}$
- Vstupná abeceda $\Sigma = \{a, b\}$
- Pásková abeceda $\Gamma = \{a, b, B\}$
- q_0 je počiatočný stav
- B je páskový symbol označujúci prázdnu bunku
- $F = \{q_f\}$
- Prechodová funkcia δ:
  $\delta(q_0, b) = (q_1, a, R), \delta(q_1, a) = (q_0, b, R), \delta(q_1, B) = (q_f, B, R)$

Úlohy:

1. Zostrojte prechodový diagram TS
2. Uveďte výpočet pre vstupy: *b*, *a*, *ba*, *bb*, *bab*
3. Pokúste sa odhadnúť, aký jazyk TS akceptuje
4. Pokúste sa zistiť, či uvedený TS zastaví pre každý možný vstupný reťazec.

Prechodový diagram:



Výpočet pre vstup b:

$$q_0 b \vdash aq_1 B \vdash aBq_f B$$

Keďže q_f je akceptačný stav, vstup b TS akceptuje.

Výpočet pre vstup a:

$$q_0 a \vdash ???$$

TS sa zastaví (zasekne) v stave q_0 pre čítaný symbol a. Keďže sa počas výpočtu TS nikdy nedostal do stavu q_f, vstup neakceptuje.

Výpočet pre vstup ba:

$$q_0 ba \vdash aq_1 a \vdash abq_0 B$$

TS sa zastaví v stave q_0 pre čítaný symbol B (t.j. hlava číta prázdnu bunku). Keďže sa počas výpočtu TS nikdy nedostal do stavu q_f, vstup neakceptuje.

Výpočet pre vstup *bb*:

$$q_0 bb \vdash aq_1 b$$

TS sa zastaví v stave q_1 pre čítaný symbol b. Keďže sa počas výpočtu TS nikdy nedostal do stavu q_f, vstup neakceptuje.

Výpočet pre vstup *bab*:

$$q_0 bab \vdash aq_1 ab \vdash abq_0 b \vdash abaq_1 B \vdash abaBq_f B$$

TS sa zastaví v stave q_f pre čítaný symbol b. Keďže sa počas výpočtu TS dostal do stavu q_f, vstup akceptuje.

Na to, aby sme popísali, aký jazyk TS akceptuje, musíme nájsť všetky reťazce, počas spracovania ktorých sa vie dostať do stavu q_f:

- Vidíme, že aby sa TS dostal do stavu q_f, musí nastať v konfigurácii situácia: $\alpha q_1 B$, kde $\alpha \in \Gamma^*$
- Keďže počas činnosti automatu sa hlava na páske posúva len doprava, znamená to, že TS musí vedieť postupne prečítať celý vstup tak, aby po poslednom čítanom symbole skončil v stave q_1.
- Vidíme, že z počiatočného stavu q_0 sa vie TS posunúť vtedy, ak aktuálne číta symbol b.
- Následne sa presúva do stavu q_1. Z tohto stavu sa vie posunúť ďalej vtedy, ak je vstup celý prečítaný, alebo sa aktuálne číta a, ktoré TS posunie naspäť do q_0.
- Čiže tie reťazce, ktoré TS vedia dostať zo počiatočného stavu q_0 do q_1 sú tie v tvare $(ba)^*b$. Preto jazyk tohto TS $L(M) = (ba)^*b$.

- Čiže tento TS akceptuje taký jazyk, ktorý je zároveň aj regulárny, čiže k nemu existuje zároveň aj konečný automat, resp. zásobníkový automat.
- To je spôsobené tým, že regulárne jazyky sú podmnožinou jazykov, ktoré vieme akceptovať TS, čiže pre každý regulárny jazyk musí existovať aj TS, ktorý ho akceptuje.
- Teraz si ešte položme otázku, či uvedený TS skončí svoju činnosť pre **akýkoľvek vstup**.

- Na začiatku je na páske uvedený nejaký vstupný reťazec $w \in \{a, b\}^*$ a číta sa jeho prvý symbol.
- Ak je prvý symbol a, TS sa zastaví, lebo nie je definovaný prechod $\delta(q_0, a)$
- Ak je prvý symbol b, TS prejde do stavu q_1 a posunie hlavu na ďalší symbol (vpravo).
- V stave q_1, ak je čítaný symbol a, TS prejde to stavu q_0 a posunie hlavu na ďalší symbol (vpravo).
- V stave q_1, ak je čítaný symbol b, TS sa zastaví, lebo nie je definovaný prechod $\delta(q_1, b)$
- Ak bol vstup celý prečítaný a TS je v stave q_0, tak sa zastaví, lebo nie je definovaný prechod $\delta(q_0, B)$
- Ak bol vstup celý prečítaný a TS je v stave q_1, tak prejde to q_f, lebo $\delta(q_1, B) = (q_f, B, R)$ a v q_f sa zastaví.

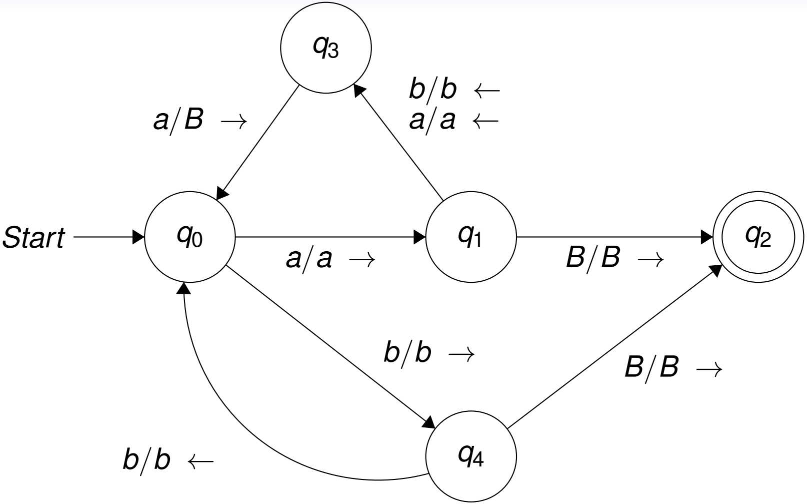
Keďže v TS sú definované operácie len tak, že sa hlava posúva vždy len vpravo na ďalší symbol, z uvedeného vyplýva, že

- Každý reťazec sa podarí dočítať do konca, t.j. hlava skončí na prázdnej bunke, ak:
  - Bol v tvare $(ba)^*b$ - v takom prípade sa TS zastaví v stave q_f.
  - Bol v tvare $(ba)^*$ - v takom prípade sa TS zastaví v stave q_0.
- Vo všetkých ostatných prípadoch sa TS zastaví predtým, než hlava prejde cez všetky symboly.
- Vo **všetkých prípadoch** však platí, že TS určite skončí svoju činnosť, t.j. neexistuje vstup, pre ktorý by sa TS "zacyklil".
- Preto tento TS naozaj spĺňa vlastnosť, že sa zastaví pre každý vstup.

Príklad č. 2

Je daný Turingov stroj prechodovým diagramom.

- Popíšte ho ako sedmicu $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ a popíšte prechodovú funkciu tabuľkou.
- Uveďte výpočet pre vstupy ab, aa, aab, abb
- Uveďte, aký jazyk uvedený TS akceptuje.
- Uveďte, či pre každý vstup platí, že sa TS zastaví počas jeho spracovania.


q_3
$b/b \leftarrow$
$a/a \leftarrow$
$a/B \rightarrow$
Start
q_0
q_1
q_2
$a/a \rightarrow$
$B/B \rightarrow$
$b/b \rightarrow$
$B/B \rightarrow$
q_4
$b/b \leftarrow$

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{a, b, B\}$
- q_0 je počiatočný stav
- $F = \{q_2\}$
- B je symbol pre prázdnu bunku
- Prechodová funkcia δ je daná tabuľkou:

Stav / Symbol	a	b	B
q_0	(q_1, a, R)	(q_4, b, R)	
q_1	(q_3, a, L)	(q_3, b, L)	(q_2, B, R)
q_2			
q_3	(q_0, B, R)		
q_4		(q_0, b, L)	(q_2, B, R)

Výpočet pre *ab*:

$$q_0 ab \vdash aq_1 b \vdash q_3 ab \vdash q_0 b \vdash bq_4 B \vdash bBq_2 B$$

Keďže sa počas výpočtu TS dostal do akceptačného stavu q_2, vstup *ab* akceptuje.

Výpočet pre *aa*:

$$q_0 aa \vdash aq_1 a \vdash q_3 aa \vdash q_0 a \vdash aq_1 B \vdash aBq_2 B$$

Keďže sa počas výpočtu TS dostal do akceptačného stavu q_2, vstup *ab* akceptuje.

Výpočet pre *aab*:

$$q_0 aab \vdash aq_1 ab \vdash q_3 aab \vdash q_0 ab \vdash aq_1 b \vdash q_3 ab \vdash q_0 b \vdash bq_4 B \vdash bBq_2 B$$

Keďže sa počas výpočtu TS dostal do akceptačného stavu q_2, vstup *ab* akceptuje.

Výpočet pre *abb*:

$$q_0abb \vdash aq_1bb \vdash q_3abb \vdash q_0bb \vdash bq_4b \vdash q_0bb \vdash bq_4b \vdash q_0bb \vdash \ldots$$

Vidíme, že nastala situácia, že sa TS zacyklil. Navyše sa nikdy nedostane do stavu q_2, preto tento vstup TS neakceptuje.

Vidíme, že sme narazili na situáciu, že sa TS zacyklí. To znamená, že existujú vstupy, pre ktoré sa nikdy nezastaví. Tieto vstupy majú vlastnosť:

- TS sa musí dostať do stavu q_4 - tam sa dostane tak, že posledný čítaný páskový symbol bolo b.
- Zároveň musí v stave q_4 znovu čítať ďalšie b, pretože toto b ho posunie do stavu q_0, pričom sa posunie hlava vľavo - t.j. na predchádzajúce b, ktoré spôsobí prechod z q_0 do q_4.
- A tieto 2 akcie sa teda môžu donekonečna opakovať.

To nám poskytuje odpoveď na otázku, či sa tento TS zacyklí pre nejaký vstup - vidíme, že áno. V tomto prípade dokonca aj vieme odhadnúť, pre aké vstupy k tomuto zacykleniu dôjde. **Toto však platí** špecificky pre tento prípad, keďže vo všeobecnosti vedieť povedať pre nejaký TS a nejaký jeho vstup, či sa zacyklí alebo zastaví je **nerozhodnuteľný problém**.

- Aby sme odhadli, aký jazyk tento TS akceptuje, musíme určiť reťazce, pre ktoré sa vie dostať do akceptačného stavu.
- Na to, aby sa TS dostal do stavu q_2 je potrebné, aby sa vedel dostať alebo do q_1, alebo do q_4 a zároveň by na páske už čítal prázdny symbol.
- Do q_4 sa vie dostať tak, aby na páske bol prázdny symbol tak, že v stave q_0 prečíta symbol b, t.j. b by bolo posledným symbolom vstupu.
- Do q_1 sa vie dostať tak, aby na páske bol prázdny symbol tak, že v stave q_1 prečíta symbol a, t.j. a by bolo posledným symbolom vstupu.

- Zo stavu q_0 sa vie do stavu q_0 vrátiť slučkou tým, že prejde cez stavy q_1, q_3. Na tento prechod je potrebné, aby v stave q_0 prečítal a a v stave q_1 čítal z pásky alebo a, alebo b, čím prejde do q_3. Pri tomto čítaní sa posúva hlava **vľavo**, teda na ten symbol a, ktorým prešiel TS zo stavu q_0 do q_1.
- Následne TS prejde do stavu q_3 do stavu q_0 tým, že toto a z pásky vymaže. De facto sa teda zo stavu q_0 vrátil do stavu q_0 tým, že z pásky prečítal a zmazal jedno a.
- Keď to dáme dokopy, zistíme, že automat dokáže akceptovať reťazce alebo v tvare a^+, alebo v tvare a^*b.
- Teda jazyk TS $L(M) = (a^*)(a + b)$

Jazyk v predchádzajúcej úlohe bol regulárny. Keďže TS vedia akceptovať nielen všetky regulárne, ale aj všetky bezkontextové jazyky, tak skúste urobiť TS, ktorý akceptuje tieto bezkontextové jazyky:

- $L_1 = \{a^n b^n c^m \mid n \geq 0, m \geq 0\}$
- $L_2 = \{w2w^R \mid w \in \{0,1\}^*\}$
- $L_3 = \{a^n b^m \mid n > m \geq 0\}$
- $L_4 = \{w \mid \sharp_b(w) = 2\sharp_a(w)\}$

TS pre jazyk L_1

Jazyk $L_1 = \{a^n b^n c^m \mid n \geq 0, m \geq 0\}$ tvoria napríklad reťazce (v zátvorke je uvedená hodnota n a m pre daný reťazec):

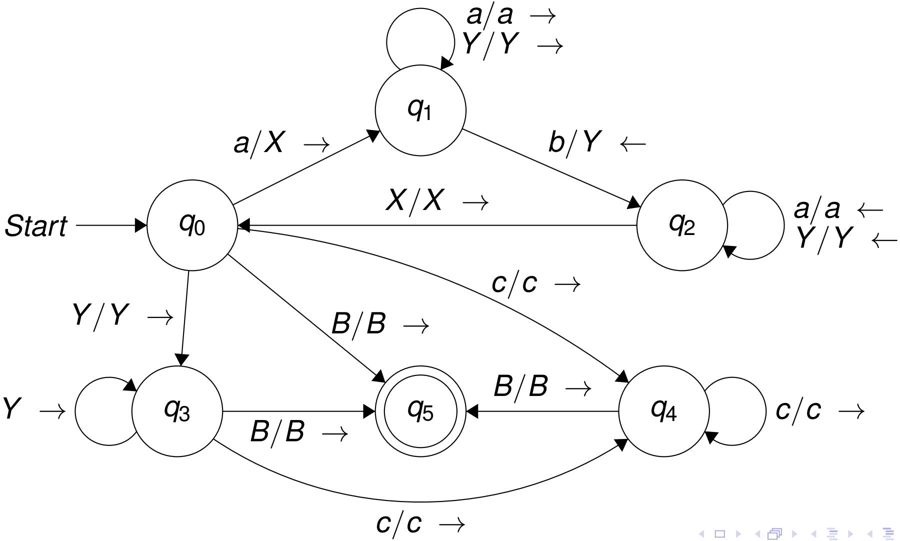
- ε $(m = n = 0)$
- ab $(n = 1, m = 0)$
- c $(n = 0, m = 1)$
- abc $(m = n = 1)$
- $aabb$ $(m = 0, n = 2)$
- cc $(m = 2, n = 0)$
- $aabbc(m = 1, n = 2)$

T.j. reťazce, kde je na začiatku postupnosť a, za ňou rovnako dlhá postupnosť b a za nimi ešte postupnosť c ľubovoľnej dĺžky

Jedným z možných spôsobov riešenia je nasledovný:

- Najprv zistíme, či reťazec začína $a^n b^n$, $n \geq 0$.
- To môžeme zistiť tak, že nájdeme prvé a, označíme si ho ako napr. X a hľadáme k nemu zodpovedajúce b, to označíme ako napr. Y.
- **Len v prípade**, že dostaneme postupnosť $X^n Y^n$, tak prefix vstupu spĺňa tvar $a^n b^n$.
- Následne môžeme zistiť, či sme práve všetko prečítali, alebo či ešte nasleduje postupnosť nejakých symbolov c, t.j. či $m = 0$ alebo $m > 0$.
- Ak áno, potom vstup spĺňa tvar $a^n b^n c^m$

Jedno z možných riešení



Vysvetlenie:

- V stave q_0 sa TS nachádza na začiatku. Ak je vstup prázdny (t.j. na vstupe je ε a na páske len prázdne symboly B), tak sa TS rovno presunie do akceptačného stavu q_5.
- Ak je na vstupe len postupnosť $c^m, m > 0$, tak sa prvým symbolom c presunie TS do stavu q_4, v tomto stave dočíta všetky c na páske a keď narazí na prvý prázdny symbol (čo nastane len v prípade, že na páske bolo $c^m, m > 0$), tak sa presunie do q_5 a akceptuje.
- Ak bol na vstupe reťazec, ktorý začínal $a^nb^n, n > 0$, tak postupné prechody q_0, q_1, q_2, q_0 párujú a a b, pričom ich prepisujú na X a Y. **Len v prípade**, že je na začiatku vstupu a^nb^n, tak TS vie prísť do q_0, na páske je $X^nY^nc^m$ a hlava sa díva na prvý symbol Y. Následne prejde do stavu q_3, v ktorom prečíta všetky Y a ak $m = 0$, prejde do q_5,ak $m > 0$, tak c-čka prečíta cez stav q_4 a prejde po ich prečítaní do q_5.

TS pre jazyk L_2

Jazyk $L_2 = \{w2w^R \mid w \in \{0,1\}^*\}$ tvoria napríklad reťazce

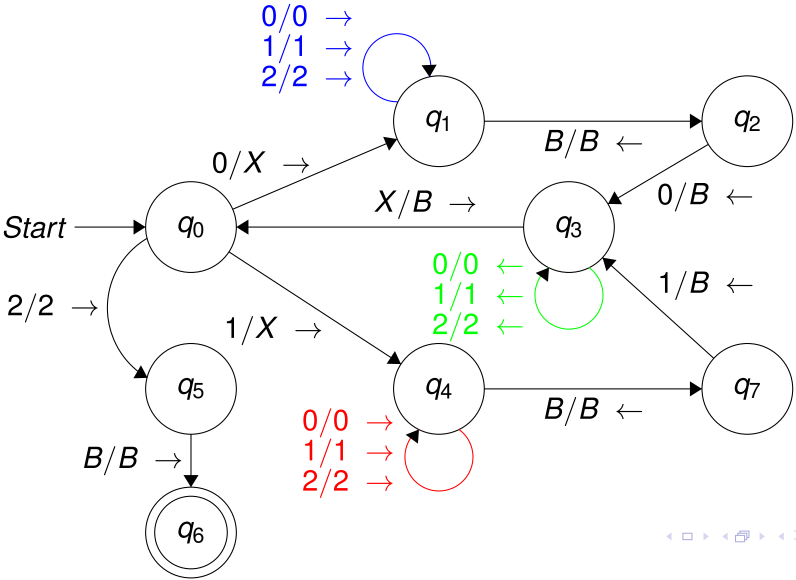
- 2
- 020
- 121
- 01210
- 10201
- 00200
- 11211

T.j. palindrómy z cifier $\{0,1,2\}$, kde cifra 2 označuje stred palindrómu, vľavo od nej je nejaký reťazec w z $\{0,1\}$ a vpravo od 2 jeho zrkadlový obraz w^R.

Jedným z možných spôsobov riešenia je nasledovný:

- Ak je na vstupe len 2, tak ju prečítame, prejdeme do iného stavu, v ktorom skontrolujeme, že už sme na páske na prázdnej bunke (tým pádom za 2 už nič nie je) a akceptujeme.
- Ak je na páske reťazec $w2w^R$, tak vezmeme prvý symbol w, zapamätáme si, čo to bolo (budeme mat rôzne stavy pre situáciu, že to bola nula alebo jednotka) a prejdeme na posledný symbol vstupu - ak súhlasí s tým, čo bolo na začiatku, tak ho zmažeme, vrátime sa na prvý symbol, zmažeme aj ten a pokračujeme s tým, čo nám zostalo. Ak by sa posledný symbol nezhodoval s prvým, tak sa zastavíme.
- Ak postupne takto mažeme prvý a posledný symbol vždy, ak sa rovnajú, tak nám na koniec zostane len 2, ktorú prečítame, skontrolujeme, či už nic na vstupe nezostalo a akceptujeme.

Jedno z možných riešení



Vysvetlenie:

- V stave q_0 sa TS nachádza na začiatku. Ak je na vstupe len 2, tak sa jej prečítaním posunie TS do q_5 a ak bola 2 jediný symbol na páske, tak sa prečítaním prázdnej bunky za 2 posunie do akceptačného q_6.
- V opačnom prípade by na vstupe malo byť $w2w^R$, kde w má aspoň 1 symbol. Ak je prvý symbol w nula, tak aj na konci w^R musí byť nula. Preto TS prejde do stavu q_1, pričom prvú nulu v w zamení za X, aby si ju označil. Následne hľadá koniec slova na páske (stav q_1, resp. prechod do q_2, keď ho nájde). V stave q_2 je na poslednom symbole slova na páske - musí to byť nula. Ak je, zmaže ju a vracia sa na začiatok - po symbol označený X. Keď ho nájde, zmaže ho a posunie sa na - teraz - prvý symbol slova na páske. Opakuje sa to, kým nezostane na páske len prostredná 2.
- Analogicky sa to deje, ak je prvý symbol jednotka - namiesto stavu q_1 a q_2 tu slúžia stavy q_4 a q_7, aby sa sme vedeli, či posledný symbol musí byť nula (q_2) alebo jednotka (q_7).

TS pre jazyk L_3

Jazyk $L_3 = \{a^n b^m \mid n > m \geq 0\}$ tvoria napríklad reťazce

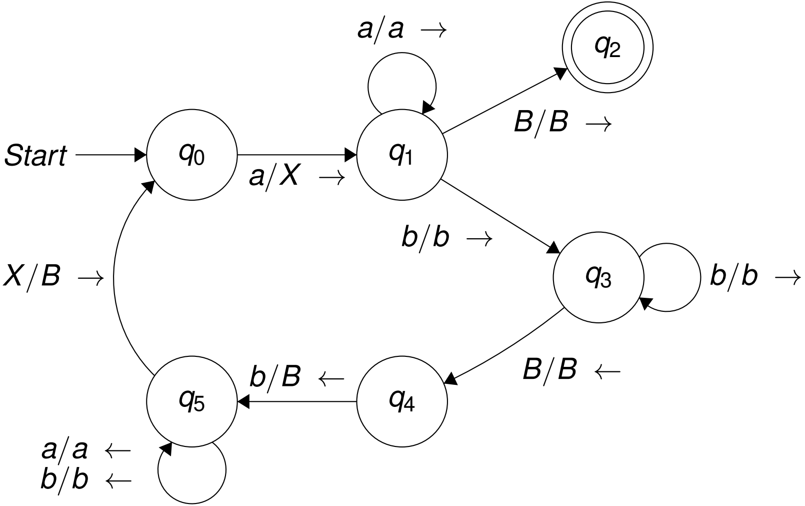
- a $(n = 1, m = 0)$
- aa $(n = 2, m = 0)$
- aab $(n = 2, m = 1)$
- $aaab$ $(n = 3, m = 1)$
- $aaabb$ $(n = 3, m = 2)$
- $aaaab$ $(n = 4, m = 1)$

T.j. reťazce, kde sú najprv a-čka, potom b-čka, pričom a-čok je ostro viacej, než b-čok.

Jedným z možných spôsobov riešenia je nasledovný:

- Vezmeme prvý symbol na páske - musí to byť *a*. Ak za ním nasleduje len postupnosť *a*-čok, t.j. $m = 0, n > m$, celý vstup spĺňa podmienku a prečítaním *a*-čok prejdeme do akceptačného stavu.
- Ak je vstup v tvare $a^n b^m$ a $n > m \geq 1$, tak sa pokúsime reťazec na páske upravovať tak, že postupne hľadáme k prvému *a* na páske posledné *b* na páske. Ak sa ich podarí nájsť, tak oba zmažeme. Opakovaním tohto postupu **musíme** z reťazca v tvare $a^n b^m, n > m$ dostať reťazec a^{n-m}, t.j. reťazec, ktorý spĺňa predchádzajúci bod.

Jedno z možných riešení



Vysvetlenie:

- Prepíšeme si prvý symbol na páske - musí ním byť a - ako X.
- Prečítaním symbolu a zároveň prejdeme do stavu q_1. V tomto stave prechádzame cez všetky a na páske. Ak sa za nimi nachádza prázdny symbol B, tak vieme, že na páske bolo $a^n b^m$, $m = 0$, t.j. akceptujeme vstup prechodom do akceptačného stavu q_2.
- Ak v stave q_1 narazíme na b, tak aby bol vstup akceptovaný, musí byť v tvare $a^n b^m$, t.j. musí nasledovať postupnosť b-čok. Snažíme sa teda prečítať všetky b (stav q_3) a prísť na ich koniec - prechod na prázdny symbol z q_3 do q_4. V stave q_4 sme teda na poslednom symbole reťazca na páske, ktorým musí byť b. To zmažeme a cez stav q_5 sa vraciame na symbol X. Keď naň narazíme, tak ho zmažeme a prepneme sa do stavu q_0.
- Tým sme vlastne z reťazca, ktorý bol na páske $a^n b^m$, urobili reťazec $a^{n-1} b^{m-1}$. Postupne toto opakujeme, kým nám na páske nezostanú len symboly a^{n-m}.
- Akýkoľvek iný reťazec by spôsobil zastavenie TS pred akceptáciou.

TS pre jazyk L_4

Jazyk $L_4 = \{w \mid \sharp_b(w) = 2\sharp_a(w)\}$ tvoria napríklad reťazce

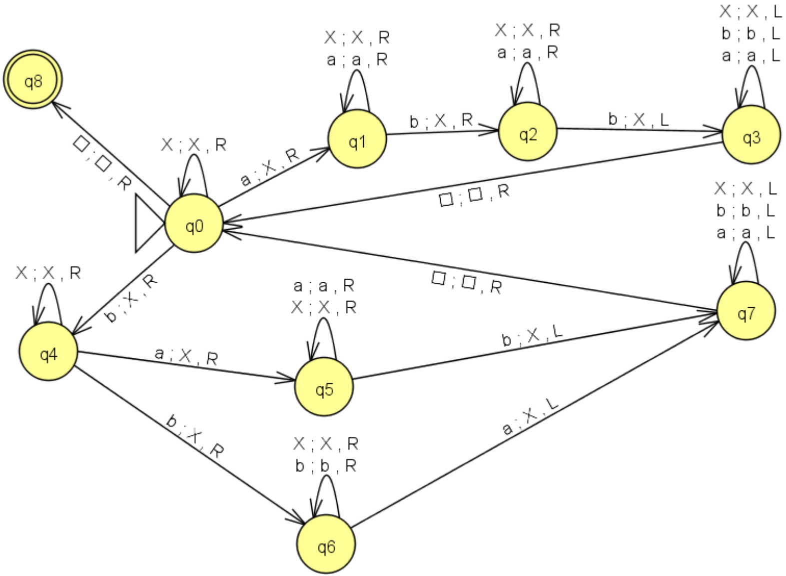
- ε (0-krát *a*, 0-krát *b*)
- *bba*, *bab*, *bba* (2-krát *b*, 1-krát *a*)
- *bbbbaa*, *bbbaba*, *bbbaab*, ..., *aabbbb* (4-krát *b*, 2-krát *a*)

T.j. reťazce, v ktorých je dvojnásobne veľa *b*-čok než *a*-čok.

Jedným z možných spôsobov riešenia je nasledovný:

- Postupne budeme prechádzať reťazec. Ak narazíme na symbol a, prepíšeme ho napríklad na X a vieme, že za ním musia niekde byť 2 b-čka. Keď ich nájdeme, obe prepíšeme na X.
- Naopak, ak narazíme na symbol b, prepíšeme ho na X a vieme, že za ním niekde musí byť jedno a a jedno b. Keď ich nájdeme, obe prepíšeme na X.
- V oboch vyššie uvedených prípadoch sa vrátime na začiatok a opakujeme.
- Ak obsahuje vstupný reťazec 2-krát viac b než a, tak po takomto prechádzaní reťazca ho nakoniec celý prepíšeme na súvislý reťazec symbolov X. Čiže ak dostaneme súvislý reťazec X-ov, tak prejdeme do akceptačného stavu.
- V opačnom prípade nám tam niekde ostane navyše a alebo b a teda vstupný reťazec akceptovať nebudeme.

Jedno z možných riešení



Predchádzajúci obrázok je z programu jFlap, na ktorý je link na webstránke. V tomto programe sa dajú kresliť aj Turingove stroje.

Rozdiel oproti našej notácii je v tom, že:

- Počiatočný stav nie je označený vstupujúcou šípkou, ale "trojuholníkom" - viď q_0.
- Prázdny symbol nie je označený B ale $\square$
- Prechod, ktorý číta symbol a, zapisuje b a hlava ide doprava (doľava) by bol u nás zapísany $a/b \rightarrow$, resp. $a/b \leftarrow$. V jFlape je zapísaný **a ; b , R**, resp. **a ; b , L**.

Vysvetlenie:

- V stave q_0 začíname čítať vstup. Ak je čítaný symbol a, znamená to, že ďalej v reťazci hľadáme 2-krát b. Prepíšeme teda toto a na X a prejdeme do stavu q_1, v ktorom hľadáme prvé b. Pri hľadaní ignorujeme iné a a ak narazíme na b, prepíšeme ho na X a prejdeme do q_2. V tomto stave znovu ignorujeme na páske a a hľadáme druhé b. Keď ho nájdeme, vraciame sa na úplný začiatok vstupu a opakujeme.
- Ak je v q_0 čítaný symbol b, znamená to, že ďalej v reťazci hľadáme 1 a a 1 b. Prepíšeme teda toto b na X a prejdeme do stavu q_4, v ktorom hľadáme a alebo b. Ak nájdeme a, prepíšeme ho na X a prejdeme do stavu q_5, v ktorom hľadáme chýbajúce b. Ak nájdeme aj to, prepíšeme ho na X a cez stav q_7 sa vraciame na začiatok reťazca. Analogicky, ak v q_4 nájdeme b, tak prejdeme do iného stavu q_6, v ktorom hľadáme chýbajúce a a po jeho nájdení znovu cez q_7 ideme naspäť do q_0.

Vysvetlenie (pokr.):

- Tým docielime po prvom prechode, že máme v reťazci 3-krát X, ktoré označí dve b a jedno a. Znovu opakujeme hľadanie ďalšej takejto trojice. Keďže teraz pri prechode reťazcom sa v ňom môžu nachádzať aj symboly X, musíme do stavov doplniť prechody, ktoré tieto X ignorujú.
- Po poslednom prechode reťazcom, ktorý posledné a a b prepíše na X sa vraciame do q_0. Ak bol reťazec z jazyka L_4, tak sa na páske nachádzajú len symboly X. V takom prípade slučka v stave q_0 zabezpečí, že sa všetky prečítajú a hlava prejde na prázdny symbol za nimi. A len v takom prípade prejdeme do akceptačného stavu q_8.
- Rovnako v prípade, že vstup je prázdny (ε), aj takýto reťazec vieme rovno akceptovať, keďže na páske sú len prázdne symboly a prejdeme z q_0 do q_8.

Okrem toho, že TS vie akceptovať všetky bezkontextové jazyky, vie akceptovať aj také jazyky, ktoré nie sú bezkontextové. Typické jazyky, ktoré nie sú bezkontextové, sú tieto 2:

- $L_5 = \{a^n b^n c^n \mid n \geq 0\}$
- $L_6 = \{ww \mid w \in \{a, b\}^*\}$

Zostrojte pre ne Turingove stroje, ktoré ich akceptujú.

TS pre jazyk L_5

Jazyk $L_5 = \{a^n b^n c^n \mid n \geq 0\}$ tvoria napríklad reťazce

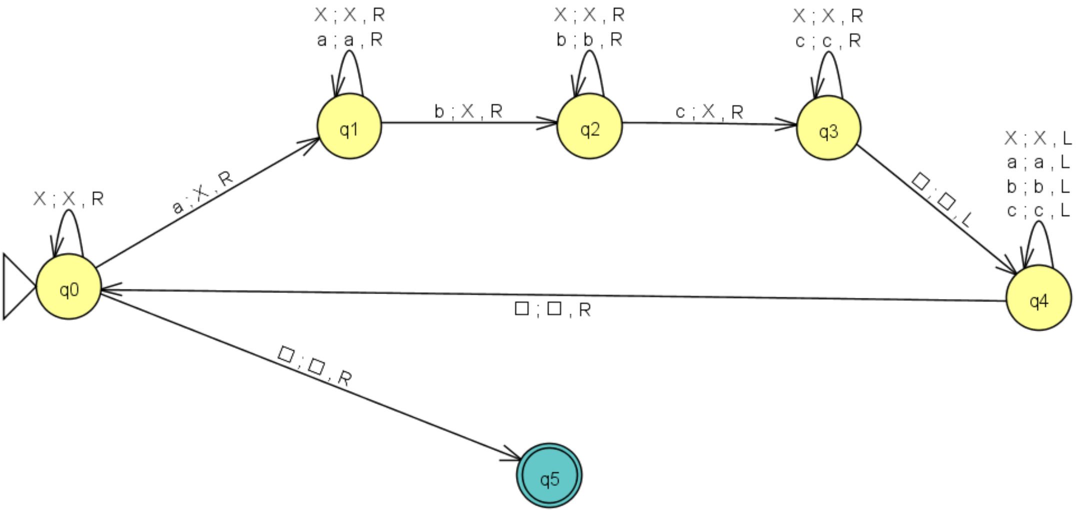
- ε $(n = 0)$
- abc $(n = 1)$
- $aabbcc$ $(n = 2)$
- $aaabbbccc$ $(n = 3)$

T.j. reťazce, kde sú najprv a-čka, potom b-čka a nakoniec$c$-čka, pričom všetkých je rovnaký počet.

Jedným z možných spôsobov riešenia je nasledovný:

- Postupne budeme prechádzať reťazec na páske a v každom prechode reťazcom si označíme jedno a, jedno b a jedno c ich prepísaním na X. Zároveň bude tento prechod taký, aby najprv hľadal a, potom b a nakoniec c.
- Ak má vstupný reťazec tvar $a^n b^n c^n$, tak po n prechodoch sa na páske objaví len postupnosť samých X (t.j. X^{3n}).
- V takom prípade vieme, že vstup môžeme akceptovať.
- Rovnako môžeme akceptovať aj prázdny vstup, t.j. ak je na páske len postupnosť prázdnych buniek.

Jedno z možných riešení



Vysvetlenie:

- Ak je na vstupe len prázdny reťazec a vtedy sa hlava díva priamo na prázdnu bunku - takom prípade priamo prejdeme do akceptačného stavu q_5.
- Ak je na vstupe $a^n b^n c^n$, tak prvý symbol musí byť a. Označíme ho prepisom na X a prejdeme do stavu q_1. V tomto stave prechádzame pásku, ignorujeme ostatné a-čka a hľadáme prvé neoznačené b (t.j. neprepísané na X). Keď ho nájdeme, prejdeme do stavu q_2.
- V stave q_2 prechádzame ostatné b-čka a hľadáme prvé neoznačené c-čko. Keď ho nájdeme, prepneme sa do stavu q_3.
- V stave q_3 prejdeme na koniec slova na páske, t.j. po prvú prázdnu bunku za slovom. Keď sa tam dostaneme, prepneme sa do q_4 a v stave q_4 sa vraciame vľavo na úplný začiatok reťazca na páske, t.j. na prázdnu bunku pred ním. Keď ho nájdeme, prepneme sa do q_0 a slovo znovu čítame od začiatku.

- Postupne tak v reťazci preznačujeme a,b,c na X. Po poslednom, n-tom prechode páskou sa znovu ocitneme v stave q_0, na prvom symbole, ktorým je teraz X. Ak sa nám podarí prejsť cez všetky symboly na páske - a všetky sú X, vieme, že vstup bol v tvare $a^n b^n c^n$ a môžeme ísť do akceptačného stavu q_5.
- Ak by vstup nebol v tvare $a^n b^n c^n$, tak sa výpočet niekde zasekne a prechod do q_5 nebude možný.
- Keďže postupne sa na páske vytvárajú symboly X, musíme ich zohľadniť v stavoch q_1, q_2, q_3, q_4 pri prechode cez symboly pásky.

TS pre jazyk L_6

Jazyk $L_6 = \{ww \mid w \in \{a, b\}^*$ tvoria napríklad reťazce

- ε
- *aa*
- *bb*
- *abab*
- *aaaa*
- *baba*
- *abbabb*

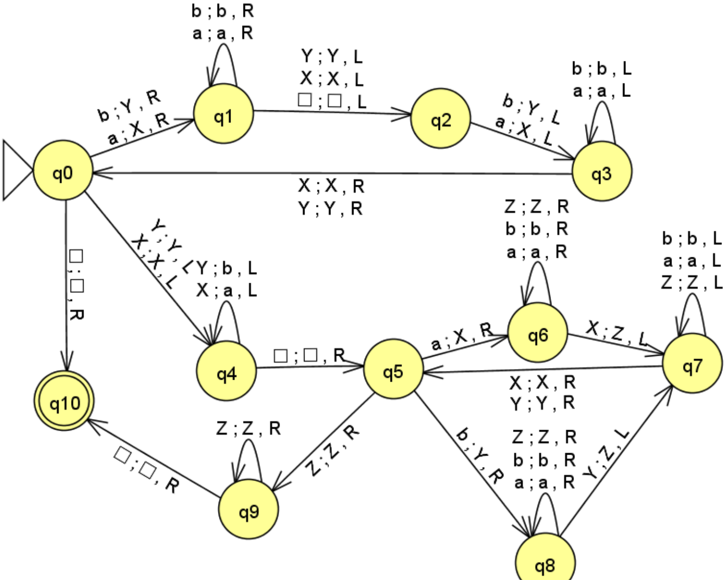
T.j. reťazce, kde pozostávajú z 2 rovnakých podreťazcov. Tento jazyk sa nazýva aj "copy"-language, lebo obsahuje 2 kópie toho istého reťazca.

Jedným z možných spôsobov riešenia je nasledovný:

- Najprv potrebujeme zistiť, kde je stred uvedeného reťazca. To môžeme vykonať tak, že vezmeme prvý symbol a prepíšeme ho na páske v štýle $a \to X, b \to Y$ a vyberieme sa na koniec reťazca, kde rovnako prepíšeme $a \to X, b \to Y$. Ak by reťazec bol *aabaab*, tak dostaneme $XabaaY$. Toto znovu zopakujeme od druhého po predposledný symbol, t.j. dostaneme $XXbaXY$. A znovu, dostaneme $XXYXXY$ s tým, že to urobíme tak, aby sa hlava nachádzala v strede slova.
- Následne sa môžeme vrátiť na začiatok, pričom cestou vrátime symboly v prvej polovici naspäť, t.j. dostaneme $aabXXY$.
- Teraz stačí postupne porovnávať, či k a v prvej časti existuje X v druhej časti, rovnako či k b existuje Y.

- Keďže teraz sú v prvej polovici reťazca symboly a, b a v druhej X, Y, vieme presne povedať, či sme v prvej alebo druhej polovici.
- Pri porovnávaní vezmeme prvý symbol (a,resp. b) a hľadáme, či prvý symbol v druhej polovici je X resp. Y. Ak áno, tak prvé a môžeme preznačiť napr. na X a spárované X napríklad na Z.
- Následne vezmeme druhý symbol v prvej polovici a porovnáme s druhým symbolom v druhej polovici, atď.
- T.j. pre reťazec na páske $aabXXY$ dostaneme $XXYZZZ$.
- Keď teda spárujeme posledný symbol z prvej polovice s posledným symbolom z druhej polovice, tak dostaneme práve reťazec, ktorý má v druhej polovici samé Z.
- Ak sa počas týchto úkonov nikde TS nezasekne skôr, než všetky X, Y v druhej časti zmení na Z, tak to znamená, že vstup bol v tvare ww, $w \in \{a, b\}^*$.

Jedno z možných riešení



Vysvetlenie:

- Ak je vstup prázdny, tak z q_0 je prechod priamo do akceptačného stavu q_{10}.
- V opačnom prípade stavy q_0, q_1, q_2, q_3 spolu spracujú reťazec na páske tak, že postupne preznačujú a na páske za X, b na páske za Y, pričom najprv spracujú prvý a posledný symbol, potom druhý a predposledný symbol, atď.
- TS vždy prečíta prvý nespracovaný symbol, t.j. prvé a, b a prejde do stavu q_1. V tomto stave sa presúva po posledný nespracovaný symbol, t.j. hľadá alebo koniec reťazca, alebo prvé X, Y, ktoré značí už-spracovaný symbol. Keď ho nájde, prejde do q_2.
- V q_2 tento krajný-pravý a, b symbol zamení za X, Y a vracia sa vľavo na prvé nespracované a, b tak, že v stave q_3 prechádza cez nespracované a, b na začiatok po posledný zatiaľ spracovaný symbol v ľavej časti, ktorý je už označený X alebo Y. Keď ho nájde, nastaví sa na prvý nespracovaný a, b symbol a prepne sa znovu do q_0 a opakuje.
- Nakoniec sa vstup upraví tak, že hlava je v stave q_0 na prvom symbole **druhej polovice** vstupu.

Vysvetlenie:

- Teraz je teda ľahko možné ísť vľavo a cestou vracať naspäť $X \to a$, $Y \to b$ (stav q_4)
- Keď nájdeme začiatok reťazca, prepneme sa do q_5. V stave q_5 máme teraz prvú polovicu reťazca zo symbolov a, b, druhú zo symbolov X, Y.
- Teraz kontrolujeme, či prvý symbol v prvej časti a, resp. b, súhlasí s prvým symbolom v druhej polovici X, resp. Y. Ak áno, tak symbol v prvej časti premenujeme na X, Y, aby sme si ho označili ako spracovaný a prislúchajúci symbol v druhej časti ako Z.
- Takto postupne spárujeme všetky symboly. Cez stav q_6 párujeme a s X, cez stav q_8 párujeme b s Y.
- Po úspešnom párovaní nastane situácia, že máme v druhej polovici len Z symboly a zároveň sme v stave q_5 a hlava je na prvom Z.
- Stačí už len dočítať Z do konca (čo sa podarí len vtedy, ak bol vstup z jazyka) a akceptovať.

Príklad algoritmu

Na TS sa vieme dívať nielen ako na zariadenie, ktoré akceptuje reťazce, ale aj ako na výpočtový model počítača. Skúste nájsť algoritmus vo forme TS, ktorý realizuje nasledovnú úlohu:

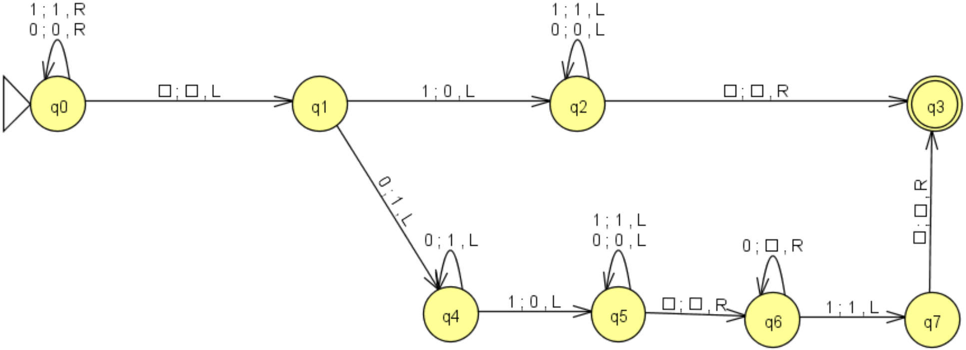
- Zostrojte TS, ktorý vezme binárne číslo a odčíta od neho jednotku. Konkrétne, na páske je zapísané binárne číslo a vašou úlohou je toto číslo prepísať na binárne číslo o 1 menšie. Zároveň nech je po skončení prepisovania čítaco-píšuca hlava na prvom symbole čísla.
- Pre jednoduchosť predpokladajte, že na číslo na páske je kladné (t.j. väčšie než nula).

Riešenie

Treba si uvedomiť, ako sa realizuje binárne odčítanie jednotky:

- Ak je posledná cifra binárneho čísla jednotka, stačí ju zmeniť na nulu, napr: 111 $\rightarrow$ 110 znamená 7 $\rightarrow$ 6
- Ak je posledná cifra binárneho čísla nula, tak ju treba zmeniť na jednotku a zároveň odčítať jednotku od "vyššieho rádu", tzv. borrow (analógia s carry operáciou pri pripočítaní jednotky): napr: 1**10** $\rightarrow$ 1**01**, t.j. 6 $\rightarrow$ 5
- T.j. ak je posledná nula, prepíšeme ju na jednotku a pozrieme sa na predchádzajúcu cifru. Ak je aj tá nula, prepíšeme na jednotku a ideme do vyššieho rádu - a tak ďalej, kým nenarazíme na jednotku, ktorú prepíšeme na nulu., napr 1000 $\rightarrow$ 0111, t.j. 8 $\rightarrow$ 7.
- Na záver môžeme zmazať nuly na najviac významných pozíciách (ak tam sú), t.j. napr. 0111 $\rightarrow$ 111.

Možné riešenie



Vysvetlenie:

- V stave q_0 hľadáme koniec vstupu - keď narazíme na prvú prázdnu bunku za vstupom, ideme do stavu q_1, hlava je na poslednom symbole vstupu.
- Ak je posledný symbol vstupu 1 - t.j. číslo končí jednotkou - zameníme ju za nulu - prechod do q_2. V q_2 sa presunieme naspäť na začiatok reťazca, t.j. kým nenarazíme na prázdnu bunku pred prvým symbolom vstupu (prechod do q_3). Následne sa posunie hlava na prvý symbol vstupu a TS vstup akceptuje a zastaví sa.
- V tomto prípade stav q_3 je akceptačný len pro forma, keďže našou úlohou nie je akceptovať reťazce, ale upraviť ich na páske.

Vysvetlenie:

- Ak je v stave q_1 posledný symbol vstupu 0 - t.j. číslo končí nulou - zameníme ju za jednotku - prechod do q_4. V stave q_4 sa potom postupne presúvame číslom na páske k vyšším rádom dovtedy, kým nuly prepisujeme na jednotky - vždy dochádza k tzv. borrow operácii. Až kým nenarazíme na jednotku - tá tam niekde byť musí, lebo predpokladáme, že číslo je väčšie ako nula. Tú jednotku prepíšeme na nulu a ideme do stavu q_5.
- V stave q_5 prejdeme na začiatok slova na páske, t.j. po prázdnu bunku pred slovom, nastavíme sa na prvý symbol vstupu a ideme do stavu q_6.
- V stave q_6 mažeme potenciálne nuly na začiatku slova na páske - aby nám zostalo slovo začínajúce jednotkou. Po nájdení vedúcej jednotky prejdeme cez stav q_7 do stavu q_3 a končíme.

Iný príklad na akceptáciu

- Zostrojte TS, ktorý akceptuje jazyk
  $L_7 = \{w\#y \mid w \in \{*\}^*, y \in \{0,1\}^*,$ y je binárna reprezentácia čísla $\sharp_*(w)\}$.
  Skúste sa zamyslieť, ako by sa pri tejto úlohe dal využiť TS z predchádzajúceho slajdu, ktorý odčituje jednotku od binárneho čísla.

TS pre jazyk L_7

Jazyk $L_7 = \{w\#y \mid w \in \{*\}^*, y \in \{0,1\}^*$, y je binárna reprezentácia čísla $\sharp_*(w)\}$ tvoria napríklad reťazce

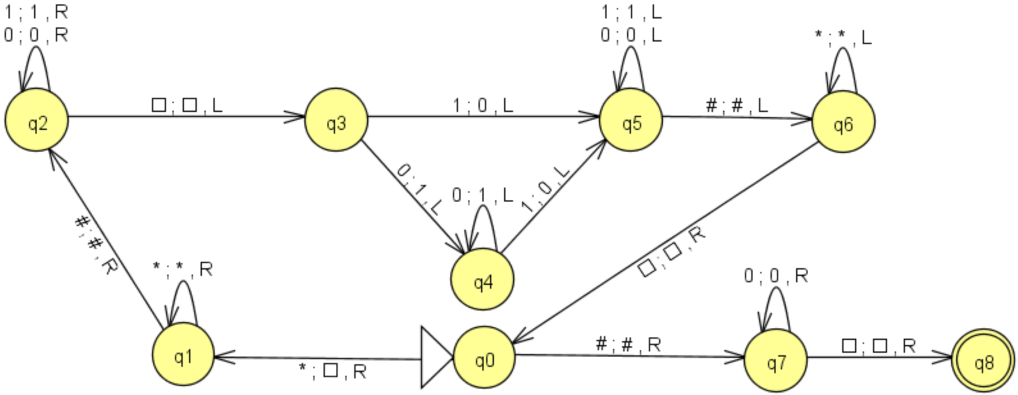
- $\#0$
- $*\#1$
- $**\#10$
- $***\#11$
- $****\#100$
- $*****\#101$
- $******\#110$

T.j. reťazce, kde pozostávajú z nejakého počtu hviezdičiek, za ktorými nasleduje mriežka, za ktorými nasleduje binárne číslo udávajúce práve počet hviezdičiek pred mriežkou.

Možné riešenie:

- Ak je vstup v požadovanom tvare, tak by malo platiť, že ak odčítame jednotku od binárneho čísla vpravo od mriežky toľkokrát, koľko hviezdičiek je vľavo od mriežky, tak tam zostane len nula.
- Čiže si môžeme pomôcť upraveným TS z predchádzajúcej úlohy.
- Postupne budeme mazať / označovať hviezdičky pred mriežkou za spracované a pre každú hviezdičku odčítame jednotku od čísla za mriežkou.
- Náš TS v predchádzajúcej úlohe predpokladal, že číslu na páske predchádza prázdny symbol - tu musíme brať do úvahy, že mu predchádza mriežka.
- Navyše náš TS v predchádzajúcej úlohe vymazal nuly zo začiatku čísla. Tu ich pre jednoduchosť môžeme ponechať, t.j. napr z čísla 1000 urobíme po odčítaní jednotky 0111.

Možné riešenie



Vysvetlenie:

- V počiatočnom stave q_0 zmažeme prvú hviezdičku, čím sa presunieme do stavu q_1. V tomto stave sa posúvame doprava na páske po mriežku. Keď ju nájdeme, presunieme sa do stavu q_2 v ktorom začína "podprogram" odčitujúci jednotku od čísla za mriežkou.
- Tento podprogram je upravená verzia TS z predchádzajúcej úlohy, ktorá predpokladá, že pred číslom je mriežka (a nie prázdna bunka) a ktorá nemaže nuly zo začiatku čísla. T.j. len sa presunie na koniec a prepíše jednotku na nulu, prípadne nuly na jednotky, kým nenájde jednotku, z ktorej urobí nulu (t.j. binárne odčítanie jednotky).
- V stave q_5 sa hlava dostane na mriežku pred číslom, v stave q_6 sa prechádza na začiatok slova na páske, t.j. na prvú hviezdičku a postup sa opakuje, t.j. znovu sa ďalšia hviezdička zmaže a znovu sa od čísla za mriežkou odčíta jednotka.

Vysvetlenie (pokr.):

- Keď sa takto spracujú všetky hviezdičky, tak nastane situácia, že TS je v stave q_0 a aktuálne sa hlava díva na mriežku. Ak bolo slovo na vstupe korektné, tak po odčítaní jednotiek musí byť za mriežkou len postupnosť núl. Preto cez stav q_7 skontrolujeme, či sú za mriežkou len nuly a ak áno, presunieme sa do akceptačného stavu q_8.
- Ak bolo za mriežkou iné číslo, než počet hviezdičiek pred mriežkou, tak:
    - Ak bolo číslo menšie než počet hviezdičiek - TS sa zasekne počas odčitovania jednotky, pretože nastane situácia, že za mriežkou je len postupnosť núl, ale TS od nej chce odčítať jednotku - čo sa nepodarí
    - Ak bolo číslo väčšie - po spracovaní všetkých hviezdičiek je toto číslo kladné, t.j. obsahuje aspoň jednu jednotku - TS sa zasekne v stave q_7 pri kontrole, či sú za mriežkou len nuly.

Ďalšie úlohy možno nájsť v knihe [1] v sekcii 8.2.7.

Použitá literatúra

1 Hopcroft, Motwani, Ullman - Introduction to Automata Theory, Languages and Computations, 3rd Ed.