Kapitola 5

Výpočtové modely a vzťahy medzi nimi

Dôvodom pre definovanie abstraktného výpočtového modelu je snaha o dokazovanie vlastností, ktoré hovoria o problémoch ako takých. Chceme hovoriť o zložitosti problémopv a tá nemá byť závislá na výbere konkrétneho počítača. Model však musí byť dostatočne silný, aby odpovedal nášmu intuitívnemu vnímaniu toho, čo je a čo nie je vypočitateľné. Zatiaľ sme mali dva modely - TS a (M)RAM.

Pre dokazovanie toho, čo sa na TS *nedá*, je jeho definícia vyhovujúca, pretože je dostatočne jednoduchá, stroj sa dá jednoducho popísať a má malú množinu elementárnych inštrukcií, ktoré môže používať. Ak chceme TS ako výpočtový model "programovať", je to nepohodlné. Zadefinujeme preto viacero jeho modifikácií, ktoré nám programovanie uľahčia. Ukážeme, že naše modifikácie nemajú vplyv na samotnú výpočtovú silu TS; len na zložitosť. Tiež si budeme všímať vzťah TS a (M)RAMu, ktorý je predsalen bližší počítaču...

Ako prvú modifikáciu dovolíme TS používať viac pások.

5.1 k-páskový Turingov stroj

Existuje viacero možností, ako definovať k-páskový TS. V podstate nám ide o to, že namiesto jednej pásky, na ktorej mohol čítať i zapisovať, uvažujeme k pások a každá má "svoju" hlavu. V jednom kroku sa TS "pozrie" na obsah políčok pod hlavami, na aktuálny stav a na základe toho zmení stav, obsah políčok pod hlavami a patrične pohne jednotlivými hlavami.

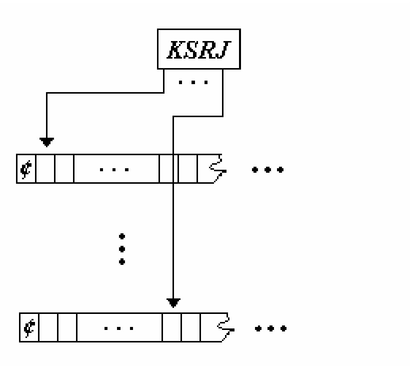
Otázkou je, kde sa nachádza vstup. Presnejšie, či ho možno prepisovať. Máme dva základné prístupy:

(i) vstupná páska sa môže prepisovať a je teda pracovná

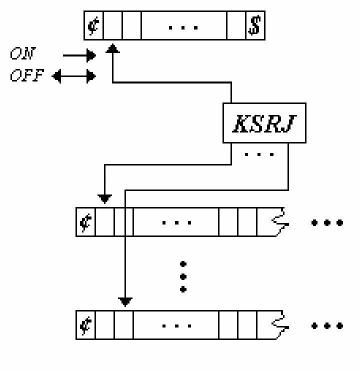
(ii) jedna z pások je len vstupom; vstup sa z nej iba číta, nesmie sa prepisovať. V tomto prípade rozpoznávame 2 modely ONLINE (jednosmerný pohyb čítacej hlavy) a OFFLINE (obojsmerný pohyb čítacej hlavy)

Vo všetkých prípadoch budeme predpokladať, že všetky pásky sú jednosmerne nekonečné smerom doprava, s ľavým okrajom označeným špeciálnym znakom ç (v prípade ONLINE/OFFLINE modelu je vstup označený aj z prava značkou \$).

Nasleduje definícia deterministického k-páskového TS, prvá páska obsahuje na začiatku výpočtu vstup.



k-páskový TS
bez oddeleného vstupu



ONLINE/ OFFLINE
k-páskový TS

Definícia 5.1 *Deterministický k-páskový TS je sedmica* $(\Sigma, \Gamma, K, \delta, B, q_0, F)$*, kde* $\Sigma, \Gamma(\Sigma \subseteq \Gamma), K, B, q_0$ *a* F *majú význam ako pri definícii TS,*

$$\delta : (\Gamma^k \times K) \rightarrowtail (\Gamma^k \times K \times \{0, 1, -1\}^k)$$

je prechodová funkcia.

Pojem konfigurácie, kroku odvodenia, výpočtu,.. sa definuje analogicky ako v prípade TS. Pritom platí:

Ak $\delta(c_1, c_2, .., c_k, k) = (c'_1, c'_2, .., c'_k, p, pos_1, .., pos_k)$
tak $\forall\ i$ také, že $c_i =$ç je $pos_i \in \{0, 1\} \wedge c'_i = c_i$. (Pripúšťame prepisovanie vstupu.)

Formálnu definíciu ONLINE,OFFLINE k-páskového TS prenechávam čitateľovi (vstupná páska sa nesmie prepisovať, pozor na pohyb hlavy na vstupe).
Základné miery zložitosti definované pre TS sú časová a pamäťová zložitosť. Definujú sa analogicky ako v prípade zložitosti konkrétnych algoritmov.

Uvedomme si, že ak uvažujeme do pamäťovej zložitosti aj priestor vstupu, je najmešia možná pamäťová zložitosť lineárna. Pritom vieme, že v prípade rozpoznávania regulárnych jazykov je vlastne pamäť konštantná. Zdá sa preto rozumné rozlišovať medzi vstupom a tým, čo naozaj musí byť v pamäti. Na úrovni abstraktného modelu to riešime takým modelom TS, ktorý má jedinú vstupnú pásku a niekoľkých pamäťových pások, ktoré možno prepisovať. Do priestorovej–pamäťovej zložitosti počítame len pamäťové pásky. Pri definovaní zložitosti uvažujeme základný model TS - deterministický viacpáskový TS s k jednosmerne nekonečnými páskami.

čas – počet krokov, resp. dĺžka výpočtu; označujeme T(n)

pamäť – maximále číslo políčka navštívené počas výpočtu na niektorej z pamäťových pások; označujeme S(n), resp. niekedy L(n)

Definícia 5.2 *Hovoríme, že*
čas *TS je* T(n) - časovo ohraničený*, ak pre žiadne vstupné slovo dĺžky n nespraví viac ako T(n) krokov; jazyk/problém je* časovej zložitosti T(n)*, ak existuje T(n)-časovo ohraničený TS, ktorý ho rozpoznáva/rieši*

priestor *TS je* S(n) -priestorovo (páskovo) ohraničený*, ak pri výpočte na žiadnom vstupnom slove dĺžky n nepoužije viac ako S(n) políčok pásky; jazyk je* priestorovej zložitosti S(n)*, ak existuje S(n) priestorovo ohraničený TS, ktorý ho rozpoznáva*

Príklad 5.1 *Napíšte TS, ktorý rozpoznáva nasledujúce jazyky*

1. $L = \{x1y|\ x, y \in \{a, b\}^*, |x| = |y|\}$
2. $L = \{wcw|\ w \in \{a, b\}^*\}$
3. $L = \{a^n b^n|\ n \geq 0\}$

Aká je zložitosť vami navrhnutých TS?

Veta 5.1 *Ku každému k-páskovému $S(n)$-páskovo ohraničenému TS existuje ekvivalentný jednopáskový $S(n)$-páskovo ohraničený TS . Ku každému k-páskovému $T(n)$-časovo ohraničenému TS existuje ekvivalentný jednopáskový $O(T^2(n))$-časovo ohraničený TS.* *k→1*

Dôkaz: : Obsahy k-pások T_k si jednopáskový T_1 uloží do stôp. Budeme mať $2k$ stôp. V každej dvojici stôp jedna stopa uchováva obsah simulovanej pásky, druhá slúži na uchovanie informácie o polohe hlavy na tejto páske. *simulácia*

Takže: v stope $2i$ bude uložený obsah i-tej pásky. Stopa $2i-1$ slúži na zapamätanie pozície hlavy na i-tej páske; j-te políčko $2i-1$ -ej stopy obsahuje špeciálny znak $\sharp$ práve vtedy ak i-ta hlava simulovaného T_k je na j-tom políčku.

Simulácia prebieha v taktoch. V jednom takte odsimuluje T_1 jeden krok T_k:

- T_1 prejde hlavou zľava doprava, pričom si do stavu zapamätá symboly $a_1, \ldots, a_k$ pod hlavami stroja T_k. V priebehu tohto posunu je T_1 v stave, ktorý vyzerá nasledovne:

  $[q, x_1, .., x_k]$, pričom

  q je stav simulovaného T_k

  $x_i = a_i$ ak v priebehu tohto posunu bola hlava T_1 na políčku, ktoré v stope $2i$ obsahovalo symbol a_i a v stope $2i-1$ bol symbol pre polohu hlavy

  $x_i = -$ ak sme v priebehu tohto posunu ešte neboli na políčku, ktoré v stope $2i-1$ obsahovalo symbol pre polohu hlavy

  Je zrejmé, že v okamihu, keď dosiahne stav $[q, a_1, .., a_k]$, má T_1 informáciu potrebnú pre "aplikovanie" prechodovej funkcie T_k.

- V stave aplikuje prechodovú funkciu. Nech $\delta_k(q, a_1, \ldots, a_k) = (p, b_1, \ldots, b_k, pos_1, \ldots, pos_k)$; T_1 si v stave uchová $(p, b_1, \ldots, b_k, pos_1, \ldots, pos_k)$

- Prechodom sprava doľava upraví pásku tak, aby odpovedala kroku T_k - prepíše a_i symbolom b_i a posunie "hlavou" j-tej pásky podľa pos_j, nastaví aktuálny stav T_k na p.

Uvedomme si, že "posun hlavy" znamená posun značky označujúcej polohu hlavy a môže znamenať, že sa T_1 posunulo vždy o jedno políčko doprava (ak sa hlava má posunúť doprava).

Teraz sa pozrime na zložitosť tejto simulácie. Nech $T_k(n), T_1(n), S_k(n), S_1(n)$ označujú časovú, resp. priestorovú zložitosť i-páskového TS $T_i, i = 1, k$. *zložitosť simulácie*

Reprezentácia viacerých pások v stopách jedinej pásky spôsobí, že priestorová zložitosť stroja je daná stopou, ktorá simuluje pôvodne najdlhšiu pásku. *pamäť*

Simulácia jedného taktu stroja T_1 je (lineárne)úmerná veľkosti pásky stroja T_k. Keďže pásková zložitosť je vždy zhora ohraničená časovou zložitosťou, dostávame: *čas*

$$T_1(n) = O(T_k(n) \times S_k(n)) = O(T_k^2(n))$$

□

Situácia je trochu iná, ak uvažujeme simuláciu k-páskového TS na dvojpáskovom TS. Možnosť používania pomocnej pásky sa odrazí na zložitosti tejto simulácie.

$k \rightarrow 2$

Veta 5.2 *Ku každému k-páskovému $T(n)$-časovo ohraničenému TS T_k existuje ekvivalentný dvojpáskový $O(T_k(n)\log T_k(n))$ časovo ohraničený TS T_2.*

Dôkaz: (Náčrt) Jednu z pások T_2 budeme používať ako pamäťovú, druhá bude pomocná. Keďže nárast času pri simulácii bol spôsobený tým, že sme v zložitosti $O(T_k(n))$ zisťovali obsah políčok pod hlavami simulovaného T_k, budeme "šetriť" tým, že po skončení každého taktu bude obsah pásky upravený tak, aby sa všetky simulované hlavy nachádzali na jednom políčku pamäťovej pásky (každá v inej stope). Popíšeme, akým spôsobom sa pomocou pomocnej pásky realizuje prepísanie čítaného symbola a posun na jednej simulovanej páske. Celkový krok T_k potom pozostáva z k po sebe nasledujúcich prepisov a posunov; zmenu na každej z k pások realizujeme zvlášť, jednu po druhej.
Pri simulácii
- predpokladáme, že páska je obojstranne nekonečná
- každá z k pások T_k je na pamäťovej páske T_2 reprezentovaná dvomi stopami. Stopy sú rozdelené (až v okamihu, keď sa na príslušné políčko dostaneme) do blokov $\ldots, B_{-i}, \ldots, B_{-1}, B_0, B_1, \ldots, B_i, \ldots$, pričom $|B_i| = |B_{-i}| = 2^{i-1}, |B_0| = 1$
- predpokladáme, že na začiatku výpočtu sú symboly uložené v spodnej stope

uloženie pásky v 2 stopách

j-ta páska simulovaného T_k je v dvoch stopách T_2 uložená nasledovne:

Pre bloky B_i a $B_{-i}, i > 0$, platí

ak je jeden plný (používa obe stopy), je druhý prázdny

oba používajú len spodnú stopu

Obsahy blokov sú po sebe idúce políčka reprezentovanej pásky, pričom

v B_i sú políčka na hornej stope pred políčkami na spodnej stope

v B_{-i} sú políčka na hornej stope pred políčkami na spodnej stope

B_i reprezentuje políčka naľavo od $B_j, 0 \leq i < j$

B_0 reprezentuje políčko pod hlavou T_k

posun hlavy vľavo

Ak simulovaný T_k chce posunúť hlavou doľava, my musíme "potiahnuť" pásku doprava. potrebujeme do bloku B_0 "dostať" prvý symbol, ktorý sa nachádza naľavo od neho. To súčasne znamená, že blok B_0 musíme uvoľniť. Budeme ho presúvať do časti napravo.
Algoritmus 12 popisuje simuláciu jedného kroku T_k. Podľa hodnoty i budeme jednému vykonaniu tohto algoritmu hovoriť B_i-operácia.

Všimnime si, že
- realizácia B_i operácie je úmerná veľkosti $|B_i|$
- po realizácii B_i-operácie používajú bloky $B_1, \ldots, B_{i-1}$ len spodnú stopu. Preto každú B_i operáciu môžme vykonať nanajvýš raz za 2^{i-1} krokov stroja T_k (musíme najprv zaplniť hornú stopu, ktorú sme pri poslednej B_i operácii vyprázdnili.)
- maximálny index i bloku, do ktorého sa počas simulácie dostane hlava T_2 je zhora ohraničený $i \leq \log T_k(n) + 1$

Algoritmus 12 posun hlavy doľava

nech $i > 0$ je najmenšie také, že blok B_i nie je plný;
if B_i nie je úplne prázdny (používa spodnú stopu) **then** prekopíruj $B_0, B_1, \ldots, B_i$ na pomocnú pásku
    ulož obsah pomocnej pásky do spodnej stopy blokov $B_1, \ldots, B_{i-1}$ a oboch stôp bloku B_i
    pomocou pomocnej pásky ulož obsah bloku B_{-i} do spodnej stopy blokov $B_{-(i-1)}, \ldots, B_0$
else (B_i je úplne prázdny) prekopíruj $B_0, B_1, \ldots, B_{i-1}$ na pomocnú pásku
    ulož obsah pomocnej pásky do spodnej stopy blokov $B_1, \ldots, B_i$
    pomocou pomocnej pásky prekopíruj obsah bloku B_{-1} do spodnej stopy blokov $B_{-i}, \ldots, B_0$

Časovú zložitosť stroja T_2 teda môžme odhadnúť nasledovne

$$T_2(n) \leq \sum_{i=1}^{\log T_k(n)+1} m \cdot 2^i \cdot \frac{T_k(n)}{2^{i-1}} \leq l \cdot T_k(n) \log T_k(n) = O(T_k(n) \log T_k(n))$$

□

5.1.1 Redukcia času a redukcia pásky

V prípade zložitosti konkrétnych algoritmov sme sa uspokojili s "rádovou" zložitosťou; používali sme asymptotiku "O", multipkikatívnu konštantu sme zanedbávali. Ako je to pri zložitosti definovanej na TS? Ukážeme, že aj v prípade TS môžme multiplikatívne konštanty zanedbať.

Veta 5.3 *Ku každému $f(n)$-priestorovo ohraničenému TS M a konštante $0 < c < 1$ existuje ekvivalentný $cf(n)$-ohraničený TS N.* *redukcia pásky*

Dôkaz: Základom dôkazu je zväčšenie abecedy páskových symbolov. Nech M je jednopáskový TS bez oddeleného vstupu, ktorého pásková zložitosť je $f(n)$; v tomto prípade zrejme $f(n) \geq n$.

$$M = (\Sigma, \Gamma, K, \delta, q_0, F)$$

Skonštruujeme nový TS N, ktorého pásková zložitosť bude nanajvýš $c \cdot f(n)$. Konštrukcia TS N je založená na tom, že jedno políčko stroja N bude simulovať m políčok stroja M. Polohu hlavy vrámci simulovanej m-tice si stroj N bude pamätať v stave ako index $i, 1 \leq i \leq m$. To umožní, aby N simuloval M krok za krokom. Presnejšie:

$$N = (\Sigma, \{B\} \cup \Gamma^m, K \cup K \times \{1, \ldots, m\} \cup K', \delta_N, q_0, F \times \{1, \ldots, m\})$$

kde prechodová funkcia δ_N je určená nie celkom formálnym popisom činnosti N.

1 TS N najprv "skomprimuje" vstup; využitím stavov z množiny K' zakóduje vstup $w = w_1 \ldots w_n$ do prvých $r, r = \lceil n/m \rceil$, políčok pásky. Nech b_i je obsah $i-$teho políčka pásky, $b_i = [w_{(i-1)m+1} w_{(i-1)m+2} \cdots a_{im}], a_j = B$ pre $j > n$. *činnosť N*

2 Počiatočnú konfiguráciu stroja M vytvorí N tak, že nastaví hlavu na prvé políčko a stav KSRJ na $[q_0, 1]$.

3 A teraz už N simuluje krok za krokom výpočet TS M. Položka i stavu určuje, kedy sa hlava simulovaného stroja hýbe v rámci snímaného políčka a nespôsobuje pohyb hlavy stroja N a kedy hlavou pohne aj N.

$\delta_N(B, [q,i]) = ([B,\dots,B],[q,i],0) \qquad\qquad q \in K$

Ku každému prvku $\boldsymbol{\delta(x,q) = (y,p,pos)}$ prechodovej funkcie stroja M vytvoríme niekoľko prvkov prechodovej funkcie stroja N. Jednotlivé prípady odpovedajú rôznym hodnotám parametra i.

$x = x_i, 1 < i < m$ $\quad \delta_N([x_1 \dots x_i \dots x_m],[q,i]) = ([x_1 \dots y \dots x_m],[p,i+pos],0)$

$x = x_i,\ i = 1$

$$\delta_N([x_1 \dots x_m],[q,1]) = \begin{cases} ([y \dots x_m],[p,1+pos],0), & pos \neq -1; \\ ([y \dots x_m],[p,m],-1), & pos = -1. \end{cases}$$

$x = x_i,\ i = m$

$$\delta_N([x_1 \dots x_m],[q,m]) = \begin{cases} ([x_1 \dots y],[p,m+pos],0), & pos \neq 1; \\ ([x_1 \dots y],[p,1],1), & pos = 1. \end{cases}$$

Aká je pamäťová zložitosť TS N? $S_N(n) = \max\{n, \lceil S(n)/m \rceil\}$. Je zrejmé, že konštantu m vieme nastaviť podľa konštanty c tak, aby $S_N(n) \leq c \cdot S(n)$. Aký má byť vzťah c a m?

Ak M je TS s oddeleným vstupom a jednou pamäťovou páskou, kumulovanie m symbolov do jediného budeme aplikovať len na pamäťovej páske. Formálny zápis nechávame ako cvičenie.

□

redukcia času

Veta 5.4 *Nech M je $f(n)$-časovo ohraničený TS, c konštanta, $0 < c < 1$, $f(n)/n \to 0$. Potom existuje ekvivalentný TS N, ktorý je $cf(n)$-časovo ohraničený.*

Dôkaz: Ak pôvodný TS M je $k-$páskový, tak ekvivalentný TS N bude $(k+1)$-páskový. Pre jednoduchosť budeme predpokladať, že M bol jednopáskový. Dôkaz pre $k > 1$ je analogický.

Z predchádzajúceho dôkazu sa necháme inšpirovať kompresiou pásky. Kompresiu využijeme k tomu, aby sme mali možnosť v jednom kroku odsimulovať m krokov výpočtu pôvodného stroja M. Uvedomme si, že k tomu, aby sme mohli odsimulovať m krokov stroja M, potrebujeme poznať nielen stav, ale aj pásku veľkosti $2m-1$, $m-1$ políčok naľavo od terajšej polohy a $m-1$ políčok napravo.

príprava simulácie

1 TS N skomprimuje vstup $a_1 \dots a_n$ do prvých $\lceil n/m \rceil$ políčok druhej pásky. Nech b_i označuje obsah $i-$teho políčka pásky, $b_i = [a_{(i-1)m+1} a_{(i-1)m+2} \cdots a_{im}]$, pričom $a_j = B$ pre $j > n$. Od tohto okamihu pásku, na ktorej bol pôvodne vstup, nepoužíva a preto ju nebudeme uvažovať.

2 TS N nastaví hlavu na prvé políčko pásky a stav KSRJ na $[q_0, 1]$; rovnako ako v predchádzajúcom dôkaze hodnota 1 indikuje, že hlava simulovaného stroja sníma prvý z m symbolov, ktoré sa nachádzajú na políčku pod hlavou.

1 takt simulácie

3 TS N pracuje **v taktoch**: nech stav N na začiatku taktu je $[q,i], 1 \leq i \leq m$, obsah snímaného políčka $a_1, \dots, a_m$

– TS N v 3 krokoch prečíta obsah domáceho aj susedných políčok; v stave si teda pamätá

$$[l_1, \dots, l_m, a_1, \dots, a_m, r_1, \dots, r_m, q, m+i]$$

– aplikovaním svojej prechodovej funkcie (v jednom kroku) v stave odsimuluje m krokov TS M - tieto sa realizujú na úseku vymedzenom domácim a susednými políčkami, preto sa to dá. Zmenený stav bude

$$[L_1, \ldots, L_m, A_1, \ldots, A_m, R_1, \ldots, R_m, p, j, k]$$

– nanajvýš v 6 krokoch (podľa info v stave) prepíše N hodnoty domácich a susedných políčok a pohne správne hlavou

Časová zložitosť stroja N je daná časom potrebným na komprimovanie vstupu a zložitosťou samotnej simulácie. *čas*

$$T_N(n) \leq 2n + 10 \cdot \lceil T_M(n)/m \rceil$$

Pri vhodnej voľbe m (akej?) môžme tvrdiť

$$2n + 10 \cdot \lceil T_M(n)/m \rceil \leq c \cdot T_M(n)$$

□

5.2 Vzťah DTS a MRAM

Ukážeme, že výpočtové modely (M)RAM a TS sú z hľadiska výpočtovej sily ekvivalentné. Sústredíme sa na zmenu zložitosti pri prechode od jedného modelu k druhému. Pri označovaní časovej zložitosti TS, resp. (M)RAMu budeme dodržiavať nasledujúce značenie

$T(n)$	časová zložitosť TS
$T_U(n)$	časová zložitosť (M)RAMu pri jednotkovej cene
$T_{\log}(n)$	časovú zložitosť (M)RAMu pri logaritmoickej cene

Veta 5.5 *Nech $M = (\Sigma, \Gamma, K, q_0, \delta, F)$ je jednopáskový TS rozhodujúci jazyk L v čase $T(n)$. Potom existuje ekvivalentý RAM, pre časovú zložitosť ktorého platí $T_U(n) = O(T(n))$, resp. $T_{\log}(n) = O(T(n) \log T(n))$* *TS ⟶ RAM*

Dôkaz: Konfiguráciu TS v RAMe[1] reprezentujeme tak, že obsah pásky uložíme do pamäte RAMu a stav si budeme pamätať návestím programu. Nezabúdajme, že pri simulovaní kroku TS potrebuje poznať nielen stav, ale aj symbol pod hlavou. Tento zistíme pomocou zapamätaného indexu registra, v ktorom sa nachádza.

Pri simulácii TS postupuje RAM nasledovne:

1 prekopíruje vstup do registrov $R(2), \ldots, R(n+1)$

2 do $R(1)$ uloží polohu hlavy na páske (číslo registra, v ktorom je uložený symbol z toho políčka pásky, na ktorom je nastavená hlava)

3 krok za krokom nasleduje simulácia TS. Pre každý stav $q \in K$ máme sadu inštrukcií, ktorá pozostáva z $|\Gamma|$ podpostupností. Nech j-ta podpostupnosť q-tej sady začína inštrukciou $N(q, j)$. Nech $\delta(\mathbf{q}, \mathbf{s_j}) = (\mathbf{p}, \mathbf{s_{j'}}, \mathbf{D})$. Potom inštrukcie j-tej podpostupnosti sú nasledujúce:

[1] RAM počíta funkciu Φ_L. Ak L je jazyk, tak Φ_L označuje charakteristickú funkciu tohto jazyka. Na slovách z jazyka dáva hodnotu 1, na slovách, ktoré nie sú z jazyka, dáva hodnotu 0.

$N(q,j)$	$LOAD*1$
$N(q,j)+1$	$SUB=j$
$N(q,j)+2$	$JZERO\ N(q,j)+4$
$N(q,j)+3$	$JUMP\ N(q,j+1)$
$N(q,j)+4$	$LOAD\ =j'$
$N(q,j)+5$	$STORE*1$
$N(q,j)+6$	$LOAD\ 1$
$N(q,j)+7$	$ADD\ =D$
$N(q,j)+8$	$STORE\ 1$
$N(q,j)+9$	$JUMP\ N(p,1)$

– simulácia začína v počiatočnom stave

4 koncovým stavom *áno, nie* odpovedajú jednoduché postupnosti inštrukcií, ktoré uložia do akumulátora 1, resp. 0 a ukončia výpočet RAMu.

Časová a pamäťová zložitosť. Uvedomme si, že uvedená simulácia platí aj pre (M)RAM. Keďže je jeden krok výpočtu TS simulovaný počtom krokov, ktoré sa dajú ohraničiť konštantou, a keďže jediný register $R(1)$ sa konštantou ohraničiť nedá ale závisí od polohy hlavy, dostávame:

TS	(M)RAM-jednotková	(M)RAM-logaritmická
T(n)	O(T(n))	O(T(n)log n)
S(n)	O(S(n))→ O(T(n))	O(S(n)log(S(n))) $\Downarrow$ O(T(n)log(T(n)))

□

Cvičenie 5.1 *Modifikujte simuláciu RAMu v prípade, že TS bol k–páskový. Zachovajte zložitosť simulácie.*

(M)RAM → TS **Veta 5.6** *Ku každému (M)RAMu existuje ekvivalentný TS.*

Dôkaz: Vychádzame z (M)RAMu, ktorý počíta funkciu $f(n)$. Konštruovaný TS bude 5-páskový. Pamäť–registre (M)RAMu budú uložené na pamäťovej páske, program spolu s čítačom inštrukcií si pamätáme v stave. Simulácia prebieha inštrukciu za inštrukciou.

1 Prvá páska obsahuje vstup

2 Druhá páska slúži na simuláciu akumulátora

3 Tretia (pamäťová) páska slúži na simuláciu pamäte (M)RAMu. Jej obsahom je postupnosť reťazcov $\sharp\sharp b(i)\sharp b(R(i))$, kde
$b(i)$ je binárne zapísaná adresa
$b(R(i))$ je binárne zapísaný obsah registra $R(i)$

4 V stave TS si pamätáme program (M)RAMu a hodnotu čítača inštrukcií

5 Simulácia (M)RAMu prebieha v taktoch; jeden takt simuluje vykonanie jednej inštrukcie (M)RAMu. Na uskutočnenie ľubovoľnej inštrukcie stačia dva operandy; tieto si môžeme uložiť na štvrtú a piatu pásku. Potom už odsimulovanie jednotlivých inštrukcií nie je problém.

Príklad simulácie $STORE * 4$

- na pamäťovej páske nájdi $\sharp\sharp b(4)\sharp b(R(4))$
- prepíš $b(R(4))$ na pomocnú pásku a nájdi na pamäťovej páske $\sharp\sharp b(R(4))\sharp b(b(R(4)))$; pamäťová páska je teda v tvare $\alpha\sharp\sharp b(R(4)\sharp b(b(R(4)))\sharp\sharp\beta$
- presuň obsah pamäťovej pásky $\sharp\sharp\beta$ na pomocnú pásku
- prekopíruj obsah akumulátora za $\alpha\sharp\sharp b(R(4)\sharp$ a na záver opäť presuň $\sharp\sharp\beta$ z pomocnej pásky

Časovú zložitosť TS získame, ak počet simulovaných inštrukcií (M)RAMu prenásobíme časom, potrebným na odsimulovanie jednej inštrukcie. Pritom *Časová zložitosť.*

- čas potrebný na simuláciu jednej RAMovskej inštrukcie je úmerný veľkosti pamäťovej pásky
- keďže násobenie a delenie sú na TS kvadratickej zložitosti, v prípade MRAMu musíme na simuláciu jednej inštrukcie uvažovať s časom kvadratickým od veľkosti pamäťovej pásky
- počet simulovaných inštrukcií je zhora ohraničený časovou zložitosťou (M)RAMu

Ostáva sa zamyslieť nad tým, *aká je veľkosť pamäťovej pásky*?

- pri logaritmickej cene $O(T(n))$ - stačí si uvedomiť, že logaritmická cena inštrukcie, ktorou (na páske vzniká, resp.) sa modifikuje ten-ktorý register, je úmerná počtu políčok, ktoré na pamäťovej páske zápis bloku zaberá
- pri jednotkovej cene - O(počet blokov $\times$ veľkosť bloku)
  počet blokov je nanajvýš $T(n)$
  veľkosť bloku ohraničíme na základe nasledujúceho faktu:

Fakt 5.7 *Nech I je vstup (M)RAMu dĺžky n, $l(B)$ maximálna dĺžka čísla použitého v programe. Po t krokoch výpočtu (M)RAMu má obsah každého registra dĺžku nanajvýš*

– $n + l(B) + t$ *v prípade RAMu;*
– c^t *v prípade MRAMu, pričom* $c = \max\{n, l(B)\}$

K dôkazu si stačí uvedomiť, že keď máme dve binárne čísla s dĺžkami binárneho zápisu a, resp. b, tak
– pre dĺžku c súčtu týchto čísel platí: $c \leq \max\{a, b\} + 1$
– pre dĺžku c súčinu týchto čísel platí: $c \leq a + b$

	časová zložitosť		TS
RAM	jednotková	$T_U(n)$	$O(T_U^3(n))$
	logaritmická	$T_{\log}(n)$	$O(T_{\log}^2(n))$
MRAM	jednotková	$T_U(n)$	$O(d^{T_U(n)})$
	logaritmická	$T_{\log}(n)$	$O(T_{\log}^3(n))$

□

5.2.1 Prvá počítačová trieda

Definícia 5.3 *Nech M, N sú dva rôzne modely. Hovoríme, že M je* polynomiálne redukovateľný *na N, ak existuje polynóm $p(n)$ taký, že ku každému výpočtu C_M na M časovej zložitosti $T_M(n)$ existuje ekvivalentný výpočet C_N na N časovej zložitosti $O(p(T_M(n)))$*

Definícia 5.4 *Modely M, N sú* polynomiálne ekvivalentné*, ak M je polynomiálne redukovateľný na N a naopak.*

Definícia 5.5 Prvú počítačovú triedu *tvoria tie výpočtové modely, ktoré sú polynomiálne ekvivalentné viacpáskovému DTS.*

Zhrnutím predchádzajúcich výsledkov dostávame.

Veta 5.8 *RAM, MRAM s logaritmickou cenou, jednopáskový a viacpáskový DTS sú polynomiálne ekvivalentné modely, ktoré patria do prvej počítačovej triedy.*