

Základy výpočtovej zložitosti

Peter Kostolányi

13. decembra 2017

Úvod

Na minulom cvičení sme sa zaoberali základmi teórie algoritmickej vypočítateľnosti – presnejšie teórie rozhodnuteľnosti – ktorej ústredným cieľom je klasifikácia výpočtových problémov podľa toho, či sú alebo nie sú algoritmicke riešiteľné. Pokročilejšie partie teórie vypočítateľnosti sa potom zaoberajú predovšetkým ďalšou klasifikáciou *neriešiteľných* problémov.

Okruh záujmu teórie výpočtovej zložitosti naopak tvoria problémy, ktoré algoritmicke riešiteľné sú. Tie sa snažia klasifikovať podľa prostriedkov, ktoré je potrebné vynaložiť na ich vyriešenie, pričom najčastejšie skúmanými mierami zložitosti sú *čas* výpočtu a *objem* pracovnej *pamäte* výpočtu.

Formalizácia intuitívnej predstavy o týchto dvoch základných mierach zložitosti sa však ukazuje omnoho problematickejšia, než tomu bolo v prípade formalizácie algoritmu, za ktorú sa všeobecne považuje na každom vstupe zastavujúci deterministický Turingov stroj. *Nie je totiž známy žiaden univerzálny spôsob*, ako výpočtovému problému priradiť jeho časovú alebo pamäťovú zložitost'. Merať časovú zložitost' problému pomocou fyzikálnej veličiny času je zjavný nezmysel, pretože takto definovaná zložitost' by nutne závisela od rýchlosti procesora a líšila by sa od počítača k počítaču (o ručnom počítaní na papieri ani nehovoriac). Pri analýze algoritmov sa teda zaužíval prístup, v ktorom sa za časovú zložitost' problému považuje počet určitých elementárnych operácií nutných na jeho vyriešenie. Napríklad pri triedeniach sa zvykne skúmať počet porovnaní, pri číselných algoritmoch počet elementárnych aritmetických operácií (sčítanie, násobenie, ...) a pod.

Na skúmanie zložitosti výpočtového problému je teda nutný *model*, ktorý určuje operácie považované za elementárne. So zmenou modelu sa môže dramaticky zmeniť aj zložitost': ak by sme napríklad pri niektorých číselných algoritmoch namiesto sčítania a násobenia považovali za elementárne iba bitové operácie, mohli by sme dospieť k drasticky odlišným výsledkom. Voľba vhodného modelu tak väčšinou závisí od konkrétnych cieľov danej analýzy.

Klasická teória výpočtovej zložitosti skúma časovú a pamäťovú náročnosť výpočtových problémov na špeciálnej triede modelov, ktorú tvoria rôzne varianty viacpáskových Turingových strojov. Za mieru času sa považuje počet krokov výpočtu Turingovho stroja, za mieru pamäte (alebo priestoru) počet zapísaných políčok na pracovných páskach. Opäť ale platí, že napríklad rôzny počet pracovných pásk stroja môže mať za následok inú časovú zložitost' problému – miery časovej zložitosti sa však obvykle definujú bez obmedzení na počet pásk.

Výhoda definície mier zložitosti pomocou Turingových strojov spočíva predovšetkým v jej exaktnosti, vďaka čomu je o časovej a priestorovej zložitosti možné dokázať aj výsledky, ktoré by pre neformálne definované miery zložitosti často nebolo možné ani sformulovať. Nevýhodou je naopak skutočnosť, že zložitost' problému na Turingovom stroji nemusí nutne zodpovedať zvyčajnej predstave o zložitosti toho istého problému.¹

Napriek tomu však existujú dobré dôvody, prečo sa zložitost'ou na Turingových strojoch zaoberať. Téma obhajujúca tento prístup je známa pod názvom *téza o invariancii* [2]. Ide, podobne ako v prípade Turingovej tézy, o neformálne a principiálne neoveriteľné tvrdenie, z ktorého vyplýva, že určité triedy zložitosti ostávajú na všetkých „rozumných“ modeloch nezmenené. Napríklad deterministický Turingov stroj s polynomiálnou časovou zložitost'ou existuje pre výpočtový problém podľa tézy o invariancii práve vtedy, keď je daný problém riešiteľný s použitím polynomiálneho počtu inštrukcií sekvenčného procesora ľubovoľnej „bežnej“ architektúry.

Práve táto nezmenenosť (invariancia) niektorých dôležitých tried časovej a priestorovej zložitosti je dôvodom, prečo je výskum v oblasti výpočtovej zložitosti zaujímavý a užitočný aj z praktického hľadiska.

¹Inými slovami: pri bežnej analýze algoritmov sa ako model väčšinou *neuvažuje* Turingov stroj, čo môže mať za následok inú výslednú zložitost' problému, než pri analýze na Turingových strojoch.

Dohoda 1. Ako model pre skúmanie deterministickej časovej zložitosti budeme v nasledujúcom *vždy* uvažovať k -páskový deterministický Turingov stroj.² Mierou časovej zložitosti bude počet krokov výpočtu takéhoto stroja. Pri priestorovej zložitosti budeme ako model uvažovať k -páskový deterministický Turingov stroj s oddelenou vstupnou páskou. Tá slúži iba na zadávanie vstupu a nie je možné meniť jej obsah. Zvyšných k pásek stroja sú bežné pracovné pásy. Mierou priestorovej zložitosti je maximálny počet použitých políčok spomedzi všetkých *pracovných* pásek.

Poznámka 1. Dôvodom, prečo sa pri priestorovej zložitosti vyžaduje oddelená vstupná páska je skutočnosť, že v opačnom prípade by priestorová zložitosť musela byť vždy aspoň taká, aká je dĺžka vstupu. Problémy so sublineárnou priestorovou zložitosťou by tak nebolo možné klasifikovať.

Definícia 1. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj. Nech $\gamma: C_0 \vdash C_1 \vdash \dots \vdash C_n$ je výpočet stroja A . Potom $\text{TIME}(A, \gamma) = n$, t.j. počet krokov výpočtu γ .

Definícia 2. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj a $w \in \Sigma^*$. Potom $\text{TIME}(A, w) = \text{TIME}(A, \gamma)$, kde γ je výpočet stroja A na slove w .

Definícia 3. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj a $n \in \mathbb{N}$. Potom

$$\text{TIME}(A, n) = \max_{|w| \leq n} \text{TIME}(A, w).$$

Definícia 4. Nech $f: \mathbb{N} \rightarrow \mathbb{N}$ je funkcia. Symbolom $\text{DTIME}(f(n))$ označujeme triedu všetkých jazykov L , pre ktoré existuje $k \in \mathbb{N}$ a k -páskový deterministický Turingov stroj A taký, že $L(A) = L$ a $\text{TIME}(A, n) = O(f(n))$.

Definícia 5. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj s oddelenou vstupnou páskou. Nech $\gamma: C_0 \vdash C_1 \vdash \dots \vdash C_n$ je výpočet stroja A . Potom $\text{SPACE}(A, \gamma)$ je maximálny počet zapísaných políčok na niektorej z pracovných pásek stroja A v niektorej z konfigurácií C_0, C_1, \dots, C_n .

Definícia 6. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj s oddelenou vstupnou páskou a $w \in \Sigma^*$. Potom $\text{SPACE}(A, w) = \text{SPACE}(A, \gamma)$, kde γ je výpočet stroja A na slove w .

Definícia 7. Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj s oddelenou vstupnou páskou a $n \in \mathbb{N}$. Potom

$$\text{SPACE}(A, n) = \max_{|w| \leq n} \text{SPACE}(A, w).$$

Definícia 8. Nech $f: \mathbb{N} \rightarrow \mathbb{N}$ je funkcia. Symbolom $\text{DSPACE}(f(n))$ označujeme triedu všetkých jazykov L , pre ktoré existuje $k \in \mathbb{N}$ a k -páskový deterministický Turingov stroj A s oddelenou vstupnou páskou taký, že $L(A) = L$ a $\text{SPACE}(A, n) = O(f(n))$.

Veta o kompresii pásky

Veta 1. Nech $f: \mathbb{N} \rightarrow \mathbb{N}$ je funkcia a $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj s oddelenou vstupnou páskou taký, že platí $\text{SPACE}(A, n) = f(n)$. Potom existuje k -páskový deterministický Turingov stroj $A' = (K', \Sigma', \Gamma', \delta', q'_0, F')$ s oddelenou vstupnou páskou taký, že $L(A') = L(A)$ a pre všetky $n \in \mathbb{N}$ je $\text{SPACE}(A', n) \leq \lceil f(n)/2 \rceil$.

Dôkaz. Idea konštrukcie spočíva v reprezentovaní dvojíc pracovných symbolov jediným symbolom z abecedy Γ^2 . To možno realizovať viacerými spôsobmi; formálne detaily jednotlivých konštrukcií prenechávame čitateľovi. \square

Poznámka 2. Iteráciou zodpovedajúcej konštrukcie možno konštantu 2 v znení predchádzajúcej vety nahradiť ľubovoľnou kladnou konštantou.

²Čitateľ by iste dokázal sformulovať definíciu takéhoto výpočtového modelu.

Poznámka 3. Je dôležité uvedomiť si, že veta o kompresii pásky je *viazaná na konkrétny model*, ktorým sú k -páskové deterministické Turingove stroje s oddelenou vstupnou páskou. Jej dôkaz využíva špeciálnu črtu Turingových strojov, ktorou je ľubovoľne veľká abeceda, pričom veľkosť abecedy na priestorovú zložitosť (z definície) nemá žiaden vplyv. Vetu preto nemožno použiť pri analýze algoritmov pre realistické modely. Napriek tomu však má veľký teoretický význam, keďže jej dôsledkom je ekvivalencia „asymptotických“ a „presných“ definícií tried priestorovej zložitosti.

Veta o lineárnom zrýchlení

Veta 2. *Nech $f: \mathbb{N} \rightarrow \mathbb{N}$ je funkcia taká, že $f(n) = \omega(n)$ a $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj taký, že $k \geq 2$ a $\text{TIME}(A, n) = f(n)$. Potom existuje $n_0 \in \mathbb{N}$ a k -páskový deterministický Turingov stroj $A' = (K', \Sigma', \Gamma', \delta', q'_0, F')$ tak, že $L(A') = L(A)$ a pre všetky $n \geq n_0$ platí $\text{TIME}(A', n) \leq \lceil f(n)/2 \rceil$.*

Dôkaz. Stroj A' bude pracovať tak, že najprv skomprimuje svoj vstup do blokov o veľkosti m , kde $m \in \mathbb{N}$ je vhodná konštanta. Pokiaľ bol pôvodný vstup na prvej páske, skomprimovaný vstup možno na druhú pásku vypísať v čase lineárnom od veľkosti vstupu, t.j. cn pre vhodné c . Stroj A' bude následne prvú pásku využívať ako pracovnú a druhá páska bude zohrávať úlohu vstupnej pásky, pričom aj na pracovných páskach bude pracovať s blokmi o veľkosti m . Bude sa teda narábať so symbolmi z abecedy Γ^m obohatenými o indikátor pozície hlavy.

Stroj A' bude zakaždým simulovať m krokov výpočtu pôvodného stroja A . To bude robiť tak, že najprv na každej páske prečíta blok, na ktorom je hlava a bloky s ním susedné, pričom si všetky tieto informácie zapamätá v stave. Táto procedúra zaberie 4 kroky. Je zrejmé, že v nasledujúcich m krokoch výpočtu pôvodného stroja sa bude pracovať iba s pozíciami v týchto blokoch. To znamená, že stroj A' vie z informácií zapamätaných v stave vypočítať nový obsah inkriminovaných troch blokov, ako aj novú pozíciu hlavy. Následné zapísanie obnovených blokov na pásku zaberie ďalšie 4 kroky. To znamená, že m krokov pôvodného stroja A dokáže stroj A' odsimulovať pomocou ôsmich krokov.

Pre čas výpočtu stroja A' potom pre každé $n \in \mathbb{N}$ platí

$$\text{TIME}(A', n) \leq c'n + \left\lceil \frac{8f(n)}{m} \right\rceil,$$

kde c' je vhodná konštanta. Je teda zrejmé, že ak $f(n) = \omega(n)$, možno zvoliť konstantu m tak, aby pre nejaké $n_0 \in \mathbb{N}$ a všetky $n \geq n_0$ platilo

$$\text{TIME}(A', n) \leq \lceil f(n)/2 \rceil,$$

čo bolo treba dokázať. □

Poznámka 4. Ako je zrejmé z dôkazu, podmienka $f(n) = \omega(n)$ je v znení vety o zrýchľovaní výpočtov kvôli kompresii vstupu, ktorá na stroji s $k \geq 2$ páskami zaberie lineárny čas. Na jednopáskovom stroji vyžaduje kompresia vstupu kvadratický čas, čo je dôvod, prečo je v znení vety podmienka $k \geq 2$.

Poznámka 5. Podobne ako pri vete o kompresii pásky, iteráciou možno konstantu 2 v znení vety nahraďiť ľubovoľnou kladnou konstantou.

Veta o redukcii počtu pásoz z k na 1 pre priestor

Veta 3. *Nech $f: \mathbb{N} \rightarrow \mathbb{N}$ je funkcia a $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -páskový deterministický Turingov stroj taký, že $\text{SPACE}(A, n) = f(n)$. Potom existuje jednopáskový deterministický Turingov stroj $A' = (K', \Sigma', \Gamma', \delta', q'_0, F')$ taký, že $L(A') = L(A)$ a*

$$\text{SPACE}(A', n) \leq f(n).$$

Dôkaz. Štandardná simulácia k -páskového deterministického Turingovho stroja na jednopáskovom s k -stopou páskou zjavne zachováva priestorovú zložitosť. □

Veta o redukcii počtu pásov z k na 1 pre čas

Veta 4. *Nech $f: \mathbb{N} \rightarrow \mathbb{N}$, $f(n) = \Omega(n)$ je funkcia a $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -pásový deterministický Turingov stroj taký, že $\text{TIME}(A, n) = f(n)$. Potom existuje jednopásový deterministický Turingov stroj $A' = (K', \Sigma', \Gamma', \delta', q'_0, F')$ taký, že $L(A') = L(A)$ a*

$$\text{TIME}(A', n) \leq c(f(n))^2,$$

kde $c \in \mathbb{R}$ je konštanta nezávislá na n .

Dôkaz. Opäť uvažujme štandardnú simuláciu k -pásového deterministického Turingovho stroja na jednopásovom s k -stopou páskou. Počet krokov potrebný na odsimulovanie jedného kroku výpočtu pôvodného stroja A je v tejto konštrukcii zjavne zhora ohraničený nejakým konštantným násobkom maximálneho počtu zapísaných políčok na niektorej z pásov stroja A . Ľahko ale vidieť, že v ľubovoľnom kroku výpočtu na vstupe dĺžky n môže byť na ľubovoľnej páske stroja A najviac $n + \text{TIME}(A, n) = O(f(n))$ zapísaných políčok. Celkovo tak simulácia zaberie najviac $O((f(n))^2)$ krokov, čo bolo treba dokázať. \square

Veta o redukcii počtu pásov z k na 2 pre čas

Veta 5 (Hennie, Stearns [1]). *Nech $f: \mathbb{N} \rightarrow \mathbb{N}$, $f(n) = \Omega(n)$ je funkcia a $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je k -pásový deterministický Turingov stroj taký, že $\text{TIME}(A, n) = f(n)$. Potom existuje dvojpásový deterministický Turingov stroj $A' = (K', \Sigma', \Gamma', \delta', q'_0, F')$ taký, že $L(A') = L(A)$ a*

$$\text{TIME}(A', n) \leq cf(n) \log_2 f(n),$$

kde $c \in \mathbb{R}$ je konštanta nezávislá na n .³

Základná myšlienka konštrukcie. Cieľom je skonštruovať stroj A' tak, aby na svojich dvoch páskach dokázal odsimulovať k -pásový stroj A s časovou zložitou zhoršenou najviac o logaritmický faktor. *Prvá páska* stroja A' bude sofistikovane štruktúrovaná a bude na nej uložený obsah všetkých k pásov stroja A . Štruktúrovanie prvej pásky má za cieľ minimalizovať priemerný počet pohybov hlavy potrebných pri simulácii jedného kroku výpočtu stroja A . *Druhá páska* stroja A' nebude mať žiadnu špeciálnu štruktúru a bude slúžiť iba ako pomocná páska pri úpravách prvej pásky.

V nasledujúcom najprv opíšeme organizáciu prvej pásky a invarianty, ktoré pre ňu budú splnené. Následne opíšeme spôsob, akým sa realizuje simulácia jedného kroku výpočtu stroja A na stroji A' a opis konštrukcie zavříme analýzou časovej zložitosti stroja A' .

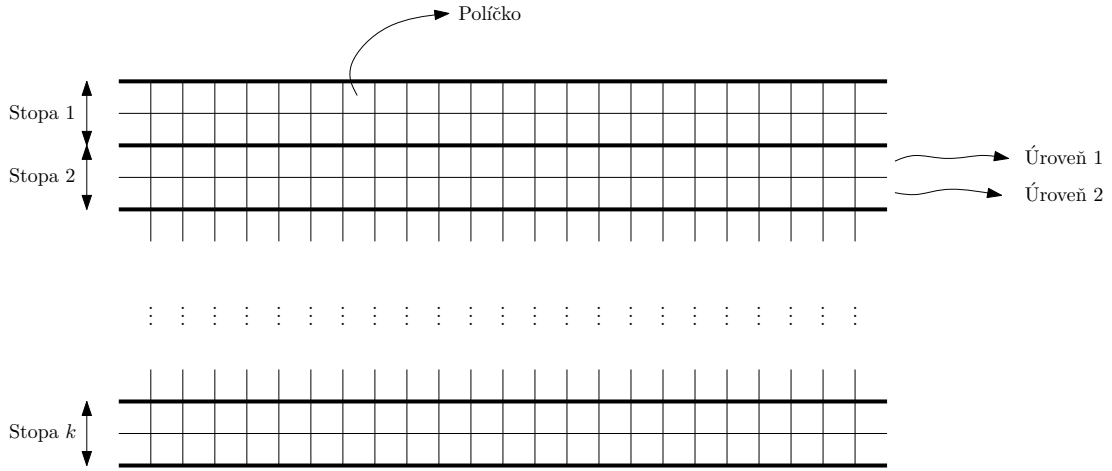
Organizácia prvej pásky. Prvú pásku stroja A' , slúžiacu na uloženie obsahov všetkých k pásov stroja A , rozdelíme na k stôp, pričom obsah každej zo stôp bude zodpovedať obsahu jednej z pásov stroja A . Každú z týchto k stôp ďalej vertikálne rozdelíme na dve úrovne. Operácie na prvej páske sa teda budú realizovať pomocou $2k$ -poschodových symbolov. Táto situácia je znázornená na obrázku 1.

Okrem opísaného vertikálneho členenia zavedieme aj horizontálne členenie pásky, ktoré ju rozdelí na bloky⁴ exponenciálne sa zväčšujúcej šírky. Jeden špeciálny stĺpec bude tvoriť samostatný blok B_0 zodpovedajúci symbolom pod čítacími hlavami pásov stroja A . Napravo od bloku B_0 sa budú nachádzať bloky B_1, B_2, B_3, \dots , pričom šírka bloku B_i bude 2^{i-1} . Podobne, naľavo od bloku B_0 sa budú nachádzať bloky $B_{-1}, B_{-2}, B_{-3}, \dots$, pričom šírka bloku B_{-i} bude 2^{i-1} . Horizontálne členenie jednej konkrétnej stopy prvej pásky je znázornené na obrázku 2.

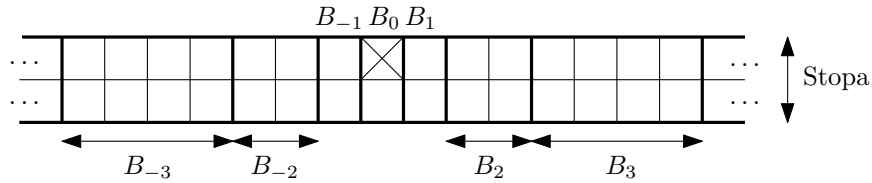
Jedna takáto dvojúrovňová stopa prvej pásky stroja A' slúži na štruktúrované uchovanie obsahu jednej pásky stroja A . Políčko (ľubovoľnej z dvoch úrovní) môže byť *plné* (ak je na ňom

³Hodnota c však závisí od k , ktoré takisto považujeme za konstantu.

⁴Takéto bloky *nebudú* (ako napríklad pri vete o kompresii pásky) realizované jedným symbolom, ale pôjde iba o virtuálne členenie stĺpcov do logických celkov.



Obr. 1: Vertikálne členenie prvej pásky stroja A' .



Obr. 2: Horizontálne členenie jednej zo stôp prvej pásky stroja A' .

zapísaný symbol z pásky stroja A) alebo *prázdne* (v opačnom prípade). Špeciálne treba upozorniť na skutočnosť, že *ak* takéto *políčko obsahuje symbol \mathbf{B} , je považované za plné*. V priebehu simulácie bude pre každý z blokov B_i platiť jedna z nasledujúcich možností:

- Blok B_i obsahuje na obidvoch vertikálnych úrovniach iba plné políčka. V takom prípade budeme hovoriť, že blok B_i je *plný*.
- Blok B_i obsahuje na obidvoch vertikálnych úrovniach iba prázdne políčka. V takom prípade budeme hovoriť, že blok B_i je *prázdny*.
- Blok B_i obsahuje na spodnej úrovni iba plné políčka a na hornej úrovni iba prázdne políčka. V takom prípade budeme hovoriť, že blok B_i je *poloprázdny*.

Počas simulácie budú navyše splnené nasledujúce invarianty:

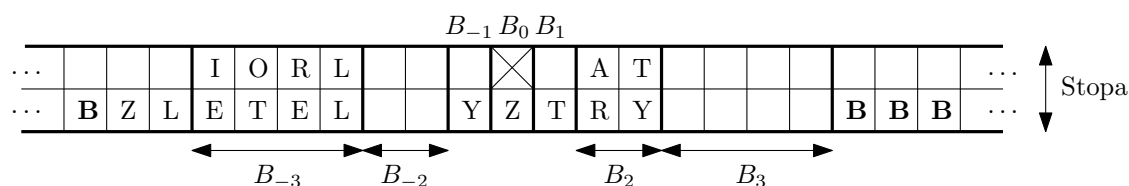
- (I1) Blok B_0 je vždy poloprázdny, pričom plné políčko na spodnej úrovni obsahuje symbol čítaný na zodpovedajúcej páske hlavou stroja A . (Z tohto dôvodu je na obrázku 2 políčko na hornej úrovni preškrtnuté.)
- (I2) Pre $i = 1, 2, \dots$ platí, že bloky B_i a B_{-i} sú buď obidva poloprázdne alebo jeden z nich je prázdny a druhý je plný.

Slovo, ktoré je obsahom s -tej pásky stroja A dostaneme z obsahu s -tej dvojúrovňovej stopy stroja A' nasledujúcim procesom:

1. Pre každý blok B_i je zodpovedajúce slovo w_i definované nasledovne:
 - Ak $i = 0$, tak w_i je symbol na spodnej úrovni bloku B_0 .

- Ak $i < 0$ a blok B_i je plný, w_i vznikne zreťazením symbolov na spodnej úrovni bloku B_i (v poradí zľava doprava) so symbolmi na hornej úrovni bloku B_i (tiež v poradí zľava doprava). Ak je blok B_i poloprázdny, w_i obsahuje iba symboly zo spodnej úrovne. Ak je blok B_i prázdny, $w_i = \varepsilon$.
 - Ak $i > 0$ a blok B_i je plný, w_i vznikne zreťazením symbolov na hornej úrovni bloku B_i (v poradí zľava doprava) so symbolmi na spodnej úrovni bloku B_i (tiež v poradí zľava doprava). Ak je blok B_i poloprázdny, w_i obsahuje iba symboly zo spodnej úrovne. Ak je blok B_i prázdny, $w_i = \varepsilon$.
2. Nech B_i je prvý a B_j je posledný blok obsahujúci nejaký symbol rôzny od **B**. Potom obsah s -tej pásky je slovo $w_i \dots w_j$ (s prípadným odignorovaním niekoľkých symbolov **B** na začiatku a na konci).

Príklad takejto reprezentácie je na obrázku 3.



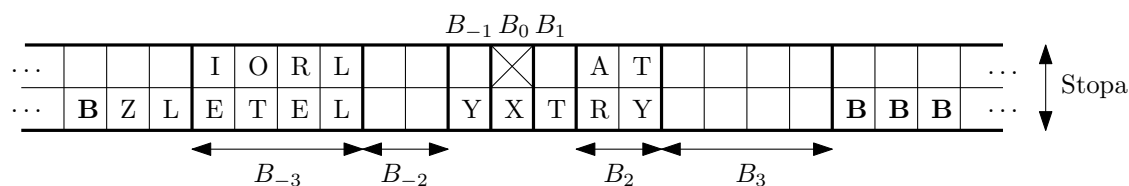
Obr. 3: Dvojúrovňová stopa prvej pásky stroja A' reprezentujúca slovo ZLETELIORLYZTATRY, pričom hlava zodpovedajúcej pásky stroja A číta druhý výskyt písmena Z.

Na začiatku simulácie sú všetky bloky všetkých stôp prvej pásky stroja A' poloprázdne. To znamená, že na spodnej úrovni prvej stopy je uložené vstupné slovo a zvyšné stopy obsahujú na spodných úrovniach symboly **B**. Horné úrovne všetkých stôp obsahujú iba prázdne políčka.

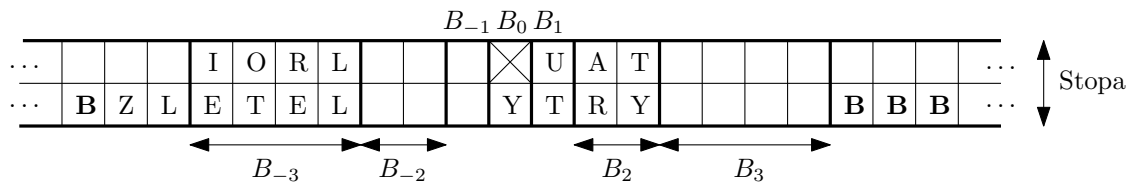
Poznamenajme ešte, že samozrejme nie je možné v konečnom čase upraviť na takýto tvar celú prvú pásku stroja A' . Simulácia preto funguje tak, že na začiatku sa na uvedený tvar upraví iba časť tejto pásky tak, aby bolo možné na prvej stope reprezentovať kompletne vstupné slovo a aby bol pre každý blok B_i na upravenej časti pásky aj blok B_{-i} . Na zvyšku pásky budú symboly **B** (stroja A' ; tieto symboly si netreba mýliť so symbolmi **B** stroja A , ktoré sa vyskytujú už v upravenej časti prvej pásky stroja A'), až kým nebude potrebné na daných pozíciách zapisovať. V takom prípade sa vytvoria nové bloky a simulácia pokračuje ďalej.

Princíp simulácie kroku výpočtu stroja A . V nasledujúcom opíšeme kľúčovú časť celej konštrukcie – spôsob, ktorým sa realizuje simulácia jedného kroku výpočtu stroja A . Táto simulácia prebieha postupne pre jednotlivé pásky stroja A , pričom vždy sa upraví zodpovedajúca stopa prvej pásky stroja A' . Keďže je procedúra pre každú z pásek stroja A rovnaká, postačí upriamiť pozornosť na jednu konkrétnu pásku stroja A a zodpovedajúcu stopu prvej pásky stroja A' .

Uvažujme napríklad situáciu na obrázku 3, kde hlava číta symbol Z. Predpokladajme, že podľa prechodovej funkcie stroja A je treba prepísať tento symbol na X, pričom hlava ostáva na mieste. Takáto situácia je jednoduchá a simulácia sa realizuje iba prepísaním daného symbolu. Výsledný obsah stopy je na obrázku 4.



Obr. 4: Situácia po simulácii kroku výpočtu, v ktorom sa Z prepísalo na X a hlava ostala na mieste.



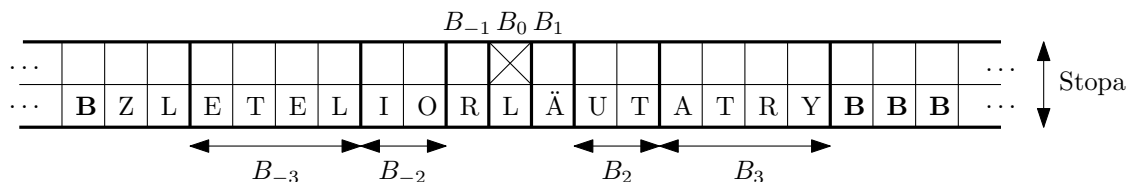
Obr. 5: Situácia po simulácii kroku výpočtu, v ktorom sa X prepísalo na U a hlava sa pohla doľava. Reprezentované slovo treba čítať ako ZLETELIORLYUTATRY.

Predpokladajme teraz, že v ďalšom kroku výpočtu sa symbol X prepíše na U a hlava sa pohne doľava. Prepísanie symbolu X na U môžeme zjavne realizovať rovnako ako v predchádzajúcom prípade. Posun hlavy však musíme realizovať posunutím obsahu pásky doprava. Vidíme ale, že na hornej úrovni bloku B_1 je voľné miesto na symbol U, ktorý tak môže uvoľniť miesto symbolu Y, ktorý posunieme z bloku B_{-1} do bloku B_0 . Výsledná situácia je na obrázku 5.

Uvažujme teraz ďalší krok výpočtu, v ktorom je treba prepísať symbol Y na symbol Ä a pohnúť hlavu znova o jedno políčko doľava. Tu narážame na problém, že na páske nie je miesto na upratanie symbolu Ä tak, aby mohol uvoľniť miesto pod čítacou hlavou ďalšiemu symbolu zľava. Táto situácia sa rieši tak, že plné bloky B_1 a B_2 sa upravujú na poloprázdne, pričom zvyšné symboly sa uložia na spodnú úroveň bloku B_3 (ak by bol blok B_3 poloprázdny, použila by sa horná úroveň bloku B_3). Všimnime si, že symboly na spodných úrovniach blokov B_1 až B_3 akurát postačujú na uskladnenie všetkých symbolov z blokov B_1 a B_2 a symbolu Ä odsunutého spod čítacej hlavy. To vysvetľuje zvolenú šírku jednotlivých blokov: pre každé $i \geq 1$ totiž platí

$$\sum_{j=1}^i 2^{j-1} = 1 + 2 \cdot \sum_{j=1}^{i-1} 2^{j-1}.$$

Pri opísanej úprave blokov B_1 , B_2 a B_3 je potrebné dbať na to, aby ostalo zachované poradie symbolov. Nakoniec treba upraviť aj bloky B_{-1} , B_{-2} a B_{-3} tak, aby bol splnený invariant (I2). To znamená presunúť hornú úroveň bloku B_{-3} na spodnú úroveň blokov B_{-2} a B_{-1} (okrem symbolu L, ktorý sa posunie pod čítaciu hlavu). Výsledná situácia je znázornená na obrázku 6.



Obr. 6: Situácia po simulácii kroku výpočtu, v ktorom sa Y prepísalo na Ä a hlava sa pohla doľava.

Myšlienka simulácie by už z uvedeného príkladu mala byť zrejmalá. Môžeme preto pristúpiť k všeobecným pravidlám pre simuláciu kroku výpočtu. Je zrejme, že postačí opísať pravidlá pre simuláciu kroku výpočtu, v ktorom sa hlava pohne doľava. Pohyb hlavy doprava sa totiž rieši symetricky a krok výpočtu bez pohybu hlavy sa ošetrí triviálnym spôsobom tak, ako je uvedené vyššie. Procedúra pre simuláciu kroku výpočtu s pohybom hlavy doľava je teda nasledovná:

1. Prepíš symbol pod čítacou hlavou (v bloku B_0) tak, ako je potrebné v simulovanom kroku výpočtu.
2. Nájdi prvý blok B_i napravo od B_0 , ktorý nie je plný.
3. Ak je blok B_i prázdny, preusporiadaj symboly v blokoch B_0, B_1, \dots, B_{i-1} tak, aby boli uložené na spodných úrovniach blokov B_1, B_2, \dots, B_i a aby reprezentované slovo ostalo nezmenené.

4. Ak je blok B_i poloprázdny, preusporiadaj symboly v blokoch B_0, B_1, \dots, B_{i-1} tak, aby boli uložené na spodných úrovniach blokov B_1, B_2, \dots, B_{i-1} a na hornej úrovni bloku B_i a aby reprezentované slovo ostalo nezmenené.
5. Nájdi prvý blok B_{-i} naľavo od B_0 , ktorý nie je prázdny.
6. Ak je blok B_{-1} plný, preusporiadaj symboly na hornej úrovni bloku B_{-i} tak, aby boli uložené na spodných úrovniach blokov $B_{-(i-1)}, B_{-(i-2)}, \dots, B_0$ a aby reprezentované slovo ostalo nezmenené.
7. Ak je blok B_{-1} poloprázdny, preusporiadaj symboly na spodnej úrovni bloku B_{-i} tak, aby boli uložené na spodných úrovniach blokov $B_{-(i-1)}, B_{-(i-2)}, \dots, B_0$ a aby reprezentované slovo ostalo nezmenené.

Všimnime si, že invarianty (I1) a (I2) ostávajú v platnosti aj po tejto transformácii.

Technické detaily simulácie kroku výpočtu stroja A . Isté technické detaily ostali nedoriešené. Napríklad je potrebné si uvedomiť, že preusporiadanie blokov v procedúre opísanej vyššie je možné urobiť v čase lineárnom od dĺžky bloku iba s použitím druhej pracovnej pásky stroja A' . Ak by bol stroj A' jednopáskový, kopírovanie by zabralo čas kvadratický od dĺžky bloku, čo by pokazilo analýzu celkovej časovej zložitosti (uvedenú nižšie).

V neformálnom opise simulácie kroku výpočtu uvedenom vyššie ďalej pracujeme s páskou na úrovni blokov. Pritom však treba mať na mysli skutočnosť, že ide iba o virtuálne bloky, z čoho vyplýva, že ich hranice je potrebné na páske označovať špeciálnymi symbolmi. Navyše, ak niekedy počas simulácie nastane situácia, že je potrebné „vytvoriť“ nový blok, je v rámci zodpovedajúcej procedúry nutné vypočítať a označiť jeho koniec. Všetky tieto detaily sa dajú ošetriť tak, aby pri simulácii nezabrali príliš veľa času. Podrobnejšie rozpracovanie takýchto úvah možno nájsť v pôvodnom článku Henniego a Stearnsa [1].

Časová zložitosť stroja A' . Pristúpme teraz k analýze časovej zložitosti stroja A' . Uvažujme najprv pevne danú pásku s stroja A .

V nasledujúcom budeme hovoriť o *kroku i -teho rádu*, ak pri simulácii daného kroku výpočtu stroja A potrebujeme v súvislosti s s -tou páskou pracovať s blokmi B_i a B_{-i} , no nepotrebujeme pracovať s blokmi B_{i+1} a $B_{-(i+1)}$. Ak implementujeme preusporiadanie blokov pomocou kopírovania na druhú pracovnú pásku stroja A , možno krok i -teho rádu zrejme odsimulovať v čase nanajvyš $c_1 \cdot 2^i$, kde c_1 je konštanta nezávislá od i . To vyplýva zo skutočnosti, že počet symbolov v prvých i blokoch je

$$\sum_{j=1}^i 2^{j-1} = O(2^i)$$

a na preusporiadanie jedného bloku stačí vďaka druhej pracovnej páske čas lineárny od dĺžky daného bloku.

Ďalej možno ľahko nahliadnuť (poriadny dôkaz je v [1]), že medzi každými dvoma krokmi rádu väčšieho ako i je nutné vykonať aspoň jeden krok rádu i . Z toho možno pomocou jednoduchšej indukcie odvodiť, že medzi každými dvoma krokmi i -teho rádu je potrebné vykonať aspoň $2^{i-1} - 1$ krokov rádu menšieho ako i . Dôsledkom je skutočnosť, že počas simulácie $T(n)$ krokov výpočtu stroja A je nanajvyš $T(n)/2^{i-2}$ týchto krokov rádu i .

Celkový počet T krokov stroja A' , zodpovedajúcich páske s a nutných na odsimulovanie $T(n)$ krokov stroja A , teda možno zhora ohraničiť nasledujúcou sumou cez rády jednotlivých krokov:

$$T \leq \sum_{i=1}^{\log T(n) + c_2} c_1 \cdot 2^i \cdot \frac{T(n)}{2^{i-2}} = c_3 \cdot T(n) \log T(n),$$

kde c_2 a c_3 sú vhodné konštanty (horná hranica sumácie je $\log T(n) + c_2$ z toho dôvodu, že počas $T(n)$ krokov je možné použiť iba $T(n)$ políčok, čomu zodpovedá $\log T(n) + c_2$ blokov).

Celková časová zložitosť stroja A' , po započítaní krokov zodpovedajúcich všetkým k páskam a lineárneho času nutného na inicializáciu výpočtu, je teda určite nanajvyš

$$c_3 \cdot k \cdot T(n) \log T(n) + O(n) \leq c \cdot T(n) \log T(n),$$

kde c je vhodná konštanta, čo bolo treba dokázať.

Literatúra

- [1] Hennie, F. C., Stearns, R. E.: Two-Tape Simulation of Multitape Turing Machines. In *Journal of the ACM*. 1966, vol. 13, no. 3, pp. 533–546.
- [2] Van Emde Boas, P.: Machine Models and Simulations. In Van Leeuwen, J. (ed.): *Handbook of Theoretical Computer Science. Volume A: Algorithms and Complexity*, pp. 1–66. Elsevier, 1990.