Cvičenie č. 12
Základy výpočtovej zložitosti

Peter Kostolányi

7. decembra 2022

1 Načo teória zložitosti založená na Turingových strojoch?

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

Okruh záujmu teórie výpočtovej zložitosti naopak tvoria problémy, ktoré algoritmicky riešiteľné sú. Tie sú v tejto teórii klasifikované podľa prostriedkov – predovšetkým *času* a objemu pracovnej *pamäte* – ktoré je potrebné vynaložiť na ich vyriešenie.

Dvoma základnými mierami zložitosti sú teda *čas* a *priestor* (alebo objem *pamäte*). Ich exaktné uchopenie sa však javí ako podstatne problematickejšie, než tomu bolo pri pojme algoritmu, za ktorého formalizáciu 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 zložitosť. Merať časovú zložitosť problému pomocou fyzikálnej veličiny času je zjavný nezmysel, pretože takto definovaná zložitosť 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, pri ktorom sa za časovú zložitosť 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 a pod.

Predpokladom skúmania zložitosti akéhokoľvek výpočtového problému je teda *model* určujúci operácie považované za elementárne. So zmenou modelu sa môže podstatne zmeniť aj zložitosť: 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.

V *teórii výpočtovej zložitosti* sa takýmto modelom najčastejšie rozumejú rôzne varianty viacpáskových *Turingových strojov*. Za mieru času sa tak považuje počet krokov výpočtu Turingovho stroja a za mieru pamäte – alebo priestoru – počet zapísaných políčok na pracovných páskach. Napríklad rôzny počet pracovných pások stroja ale opäť môže mať za následok inú časovú zložitosť problému.[1]

Prednosťou takéhoto prístupu k skúmaniu výpočtovej zložitosti je predovšetkým jeho exaktnosť, vďaka ktorej je o časovej a priestorovej zložitosti možné dokázať aj výsledky, ktoré by pre čisto intuitívne chápané miery zložitosti často nebolo možné ani sformulovať. Jeho nevýhodou je naopak skutočnosť, že zložitosť problému na Turingovom stroji nemusí nutne zodpovedať zvyčajnej predstave o zložitosti toho istého problému.[2]

Napriek tomu však existujú dobré dôvody, prečo sa teóriou zložitosti založenou na Turingových strojoch zaoberať. Azda najpodstatnejší z nich je vyjadrený tzv. *tézou o invariancii* [2]. Ide, podobne ako v prípade Turingovej tézy, o principiálne neoveriteľné tvrdenie, podľa ktorého určité triedy zložitosti ostávajú pre všetky „rozumné“ modely nezmenené. Napríklad deterministický Turingov stroj s polynomiálnou časovou zložitosťou existuje pre výpočtový problém práve vtedy, keď je tento problém riešiteľný s použitím polynomiálneho počtu inštrukcií sekvenčného procesora ľubovoľnej „bežnej“ architektúry.

Práve táto nezmenenosť – alebo invariancia – niektorých dôležitých tried časovej a priestorovej zložitosti pri zmene uvažovaného modelu je dôvodom, prečo je výskum v oblasti výpočtovej zložitosti zaujímavý a užitočný aj z praktického hľadiska.

[1] Miery časovej zložitosti sa obvykle definujú bez obmedzení na počet pások.

[2] Pri bežnej analýze algoritmov sa ako model väčšinou *neuvažuje* Turingov stroj, čo môže mať za následok inú výslednú zložitosť problému, než pri analýze na Turingových strojoch.

2 Základné definície

Dohoda 1. V nasledujúcom pracujeme iba s deterministickými Turingovými strojmi, ktoré sa na každom vstupe zastavia – výpočet stroja na každom slove je teda vždy konečný. Tento predpoklad už ďalej uvádzať nebudeme.

Dohoda 2. Ako model pre skúmanie deterministickej časovej zložitosti budeme v nasledujúcom *vždy* uvažovať *viacpáskový deterministický Turingov stroj* – čiže *k-páskový deterministický Turingov stroj* pre nejaké $k \in \mathbb{N} - \{0\}$.[3] Mierou časovej zložitosti bude počet krokov výpočtu takéhoto stroja. Pri priestorovej zložitosti budeme ako model uvažovať *viacpáskový deterministický Turingov stroj s oddelenou vstupnou páskou*. Tá slúži iba na zadávanie vstupu a nie je možné meniť jej obsah. Zvyšných $k \in \mathbb{N} - \{0\}$ pások stroja sú bežné pracovné pásky. Mierou priestorovej zložitosti je maximálny počet použitých políčok spomedzi všetkých *pracovných* pások.

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 $k \in \mathbb{N} - \{0\}$ a A je k-páskový deterministický Turingov stroj. Nech

$$\gamma\colon C_0 \vdash C_1 \vdash \ldots \vdash C_n,$$

kde $C_0, \ldots, C_n$ sú konfigurácie, je výpočet stroja A. Potom $\mathrm{TIME}(A, \gamma) = n$; ide teda o počet krokov výpočtu tvoriacich výpočet γ.

Definícia 2. Nech $k \in \mathbb{N} - \{0\}$, A je k-páskový deterministický Turingov stroj a $w \in \Sigma^*$. Potom $\mathrm{TIME}(A, w) = \mathrm{TIME}(A, \gamma)$, kde γ je výpočet stroja A na slove w.

Definícia 3. Nech $k \in \mathbb{N} - \{0\}$, A je k-páskový deterministický Turingov stroj a $n \in \mathbb{N}$. Potom

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

Definícia 4. Nech $f\colon \mathbb{N} \to \mathbb{N}$ je funkcia. Trieda $\mathrm{DTIME}(f(n))$ je tvorená všetkými jazykmi L, pre ktoré existuje $k \in \mathbb{N} - \{0\}$ a k-páskový deterministický Turingov stroj A taký, že $L(A) = L$ a $\mathrm{TIME}(A, n) = O(f(n))$.

Definícia 5. Nech $k \in \mathbb{N} - \{0\}$ a A je k-páskový deterministický Turingov stroj s oddelenou vstupnou páskou. Nech

$$\gamma\colon C_0 \vdash C_1 \vdash \ldots \vdash C_n,$$

kde $C_0, \ldots, C_n$ sú konfigurácie, je výpočet stroja A. Potom $\mathrm{SPACE}(A, \gamma)$ je maximálny počet zapísaných políčok na niektorej z pracovných pások stroja A v niektorej z konfigurácií $C_0, C_1, \ldots, C_n$.

Definícia 6. Nech $k \in \mathbb{N} - \{0\}$, A je k-páskový deterministický Turingov stroj s oddelenou vstupnou páskou a $w \in \Sigma^*$. Potom $\mathrm{SPACE}(A, w) = \mathrm{SPACE}(A, \gamma)$, kde γ je výpočet stroja A na slove w.

Definícia 7. Nech $k \in \mathbb{N} - \{0\}$, A je k-páskový deterministický Turingov stroj s oddelenou vstupnou páskou a $n \in \mathbb{N}$. Potom

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

Definícia 8. Nech $f\colon \mathbb{N} \to \mathbb{N}$ je funkcia. Trieda $\mathrm{DSPACE}(f(n))$ je tvorená všetkými jazykmi L, pre ktoré existuje $k \in \mathbb{N} - \{0\}$ a k-páskový deterministický Turingov stroj A s oddelenou vstupnou páskou taký, že $L(A) = L$ a $\mathrm{SPACE}(A, n) = O(f(n))$.

[3] Čitateľ by iste dokázal sformulovať definíciu takéhoto výpočtového modelu.

3 Veta o kompresii pásky

Veta 1. *Nech* $f : \mathbb{N} \to \mathbb{N}$ *je funkcia,* $k \in \mathbb{N} - \{0\}$ *a* A *je* k*-páskový deterministický Turingov stroj s oddelenou vstupnou páskou taký, že* $\mathrm{SPACE}(A, n) = f(n)$*. Potom existuje* k*-páskový deterministický Turingov stroj* A' *s oddelenou vstupnou páskou taký, že* $L(A') = L(A)$ *a pre všetky* $n \in \mathbb{N}$ *je* $\mathrm{SPACE}(A', n) \leq \lceil f(n)/2 \rceil$.

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. □

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

Poznámka 1. Veta o kompresii pásky je *viazaná na konkrétny model*, ktorým sú k-páskové deterministické Turingove stroje s oddelenou vstupnou páskou. Jej dôkaz totiž využíva špeciálnu črtu Turingových strojov, ktorou je ľubovoľne veľká pracovná abeceda – veľkosť tejto abecedy pritom na priestorovú zložitosť (z definície) nemá žiaden vplyv. Vetu preto napríklad nemožno aplikovať v realistickejšej situácii, keď je potrebné odhadnúť potrebný počet *bitov* pamäte. Napriek tomu však má veľký teoretický význam – vďaka nej by sme napríklad v definícii 8 mohli nahradiť podmienku $\mathrm{SPACE}(A, n) = O(f(n))$ o niečo silnejšou požiadavkou $\mathrm{SPACE}(A, n) \leq f(n)$ pre všetky dostatočne veľké $n \in \mathbb{N}$.

4 Veta o lineárnom zrýchlení

Veta 2. *Nech* $f : \mathbb{N} \to \mathbb{N}$ *je funkcia taká, že* $f(n) = \omega(n)$*, nech* $k \geq 2$ *je prirodzené číslo a nech* A *je* k*-páskový deterministický Turingov stroj taký, že* $\mathrm{TIME}(A, n) = f(n)$*. Potom existuje* $n_0 \in \mathbb{N}$ *a* k*-páskový deterministický Turingov stroj* A' *taký, že* $L(A') = L(A)$ *a pre všetky* $n \geq n_0$ *je* $\mathrm{TIME}(A', n) \leq \lceil f(n)/2 \rceil$.

Dôkaz. Stroj A' najprv skomprimuje svoj vstup do blokov o veľkosti m, kde $m \in \mathbb{N} - \{0\}$ 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 simulovať výpočet pôvodného stroja A po úsekoch pozostávajúcich z m krokov výpočtu stroja A. V rámci simulácie každého takéhoto úseku najprv na každej páske prečíta blok čítaný hlavou 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 políčkami 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

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

kde c' je vhodná konštanta. Vďaka predpokladu $f(n) = \omega(n)$ teda evidentne možno zvoliť konštantu m tak, aby pre nejaké $n_0 \in \mathbb{N}$ a všetky $n \geq n_0$ skutočne bolo

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

Ako je zrejmé z dôkazu, podmienka $f(n) = \omega(n)$ je v znení predchádzajúcej vety 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 $k \geq 2$.

Podobne ako pri vete o kompresii pásky, iteráciou možno konštantu 2 v znení vety nahradiť ľubovoľnou kladnou konštantou.

5 Veta o redukcii počtu pások z k na 1 pre priestor

Veta 3. *Nech $f\colon \mathbb{N} \to \mathbb{N}$ je funkcia, $k \in \mathbb{N} - \{0\}$ a A je k-páskový deterministický Turingov stroj s oddelenou vstupnou páskou taký, že $\mathrm{SPACE}(A, n) = f(n)$. Potom existuje jednopáskový deterministický Turingov stroj A' s oddelenou vstupnou páskou taký, že $L(A') = L(A)$ a pre všetky $n \in \mathbb{N}$ je*

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

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

6 Veta o redukcii počtu pások z k na 1 pre čas

Veta 4. *Nech $f\colon \mathbb{N} \to \mathbb{N}$ je funkcia spĺňajúca $f(n) = \Omega(n)$, nech $k \in \mathbb{N} - \{0\}$ a A je k-páskový deterministický Turingov stroj taký, že $\mathrm{TIME}(A, n) = f(n)$. Potom existuje $n_0 \in \mathbb{N}$ a jednopáskový deterministický Turingov stroj A' taký, že $L(A') = L(A)$ a pre všetky $n \geq n_0$ je*

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

kde $c > 0$ je reálna konštanta nezávislá na n.

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

7 Veta o redukcii počtu pások z k na 2 pre čas

Veta 5 (Hennie, Stearns [1]). *Nech $f\colon \mathbb{N} \to \mathbb{N}$ je funkcia spĺňajúca $f(n) = \Omega(n)$, nech $k \in \mathbb{N} - \{0\}$ a A je k-páskový deterministický Turingov stroj taký, že $\mathrm{TIME}(A, n) = f(n)$. Potom existuje $n_0 \in \mathbb{N}$ a dvojpáskový deterministický Turingov stroj A' taký, že $L(A') = L(A)$ a pre všetky $n \geq n_0$ je*

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

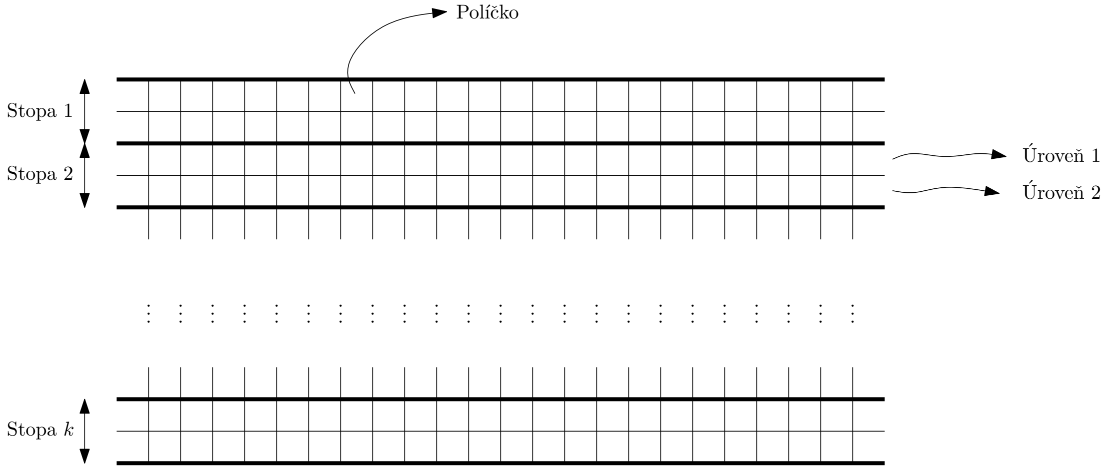
kde $c > 0$ je reálna konštanta nezávislá na n.[4]

Základná myšlienka konštrukcie. Cieľom je skonštruovať stroj A' tak, aby na svojich dvoch páskach dokázal odsimulovať k-páskový stroj A s časovou zložitosťou zhoršenou nanajvýš o logaritmický faktor. *Prvá páska* stroja A' bude sofistikovane štruktúrovaná a bude na nej uložený obsah všetkých k pások 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 zavŕšime analýzou časovej zložitosti stroja A'.

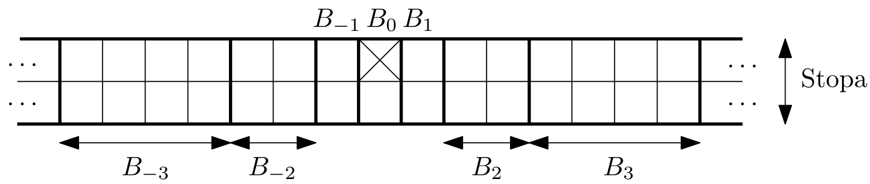
[4] Hodnota c však závisí od k, ktoré takisto považujeme za konštantu.

Organizácia prvej pásky. Prvú pásku stroja A', slúžiacu na uloženie obsahov všetkých k pások stroja A, rozdelíme na k stôp, pričom obsah každej z nich bude zodpovedať obsahu jednej z pások stroja A. Každú z týchto k stôp ďalej vertikálne rozdelíme na dve úrovne. Slová zapísané na prvej páske teda budú pozostávať z vhodných $2k$-poschodových symbolov. Táto situácia je znázornená na obrázku 1.



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

Okrem opísaného vertikálneho členenia zavedieme aj horizontálne členenie pásky, ktoré ju rozdelí na bloky[5] exponenciálne sa zväčšujúcej šírky. Jeden špeciálny stĺpec bude tvoriť samostatný blok B_0 zodpovedajúci symbolom pod čítacími hlavami pások stroja A. Napravo od bloku B_0 sa budú nachádzať bloky $B_1, B_2, B_3, \ldots$, pričom šírka bloku B_i bude 2^{i-1}. Podobne, naľavo od bloku B_0 sa budú nachádzať bloky $B_{-1}, B_{-2}, B_{-3}, \ldots$, pričom šírka bloku B_{-i} bude 2^{i-1}. Horizontálne členenie jednej konkrétnej stopy prvej pásky je znázornené na obrázku 2.



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

Jedna takáto dvojúrovňová stopa prvej pásky stroja A' slúži na štruktúrované uchovanie obsahu jednej pásky stroja A. Políčko ľubovoľnej z oboch úrovní môže byť *plné* (ak je na ňom zapísaný symbol z pásky stroja A) alebo *prázdne* (v opačnom prípade). Špeciálne treba upozorniť na skutočnosť, že *ak* takéto *políčko obsahuje symbol* **B**, *je považované za plné.* V priebehu simulácie bude mať každý z blokov B_i jednu z nasledujúcich vlastností:

- 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*.

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

Počas simulácie budú navyše splnené nasledujúce invarianty:

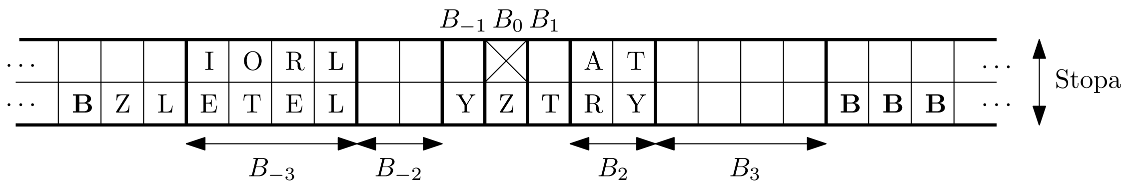
($I1$) Blok B_0 je vždy poloprázdny, pričom plné políčko na spodnej úrovni obsahuje symbol čítaný na príslušnej páske hlavou stroja A. (Z tohto dôvodu je na obrázku 2 políčko na hornej úrovni preškrtnuté.)

($I2$) Pre $i = 1, 2, \ldots$ 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 prvej pásky 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 $w_i \ldots w_j$ (s prípadným odignorovaním niekoľkých symbolov **B** na začiatku a na konci).

Príklad takejto reprezentácie je na obrázku 3.



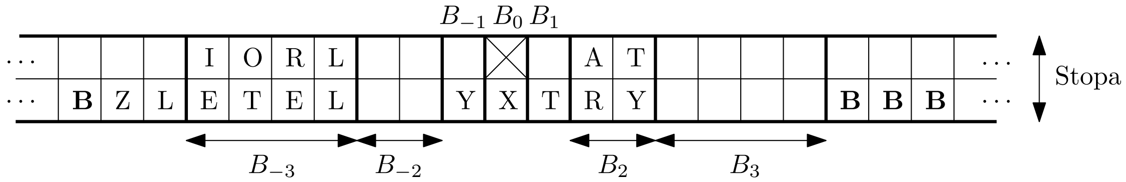
Obr. 3: Dvojúrovňová stopa prvej pásky stroja A' reprezentujúca slovo `ZLETELIORLYZTATRY`, pričom hlava zodpovedajúcej pásky stroja A číta druhý výskyt písmena `Z`.

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.

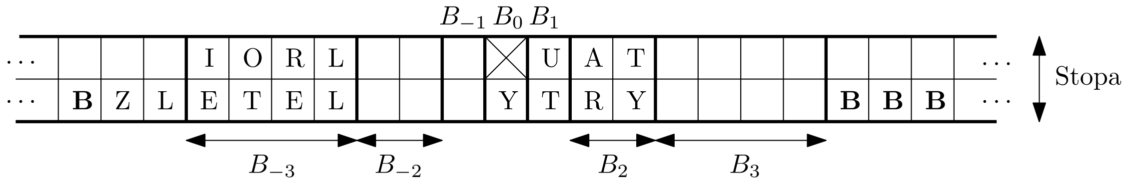
Poznamenajme ešte, že samozrejme nie je možné v konečnom čase upraviť na takýto tvar celú prvú pásku stroja A'. Na začiatku simulácie sa preto na uvedený tvar upraví iba časť tejto pásky tak, aby bolo možné na prvej stope reprezentovať kompletné vstupné slovo a aby bol pre každý blok B_i na upravenej 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 pások 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.



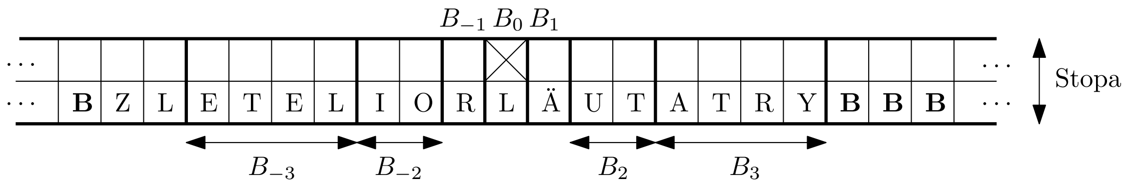
Obr. 5: Situácia po simulácii kroku výpočtu, v ktorom sa `X` prepísalo na `U` a hlava sa pohla doľava. Reprezentované slovo treba čítať ako `ZLETELIORLYUTATRY`.

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 pre symbol `U`, ktorý tak môže uvoľniť miesto symbolu `Y` – ten presunieme 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: 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 upravia na poloprázdne, pričom zvyšné symboly sa uložia na spodnú úroveň bloku B_3; keby 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

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

Pri opísanej úprave blokov B_1, B_2 a B_3 je potrebné myslieť na to, aby zostalo 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 spodné úrovne 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 zrejmá. Môžeme preto pristúpiť k všeobecnému opisu pravidiel pre simuláciu jedného kroku výpočtu. Je zrejmé, 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 $B_0, B_1, \ldots, B_{i-1}$ tak, aby boli uložené na spodných úrovniach blokov $B_1, B_2, \ldots, B_i$ a aby reprezentované slovo ostalo nezmenené.
4. Ak je blok B_i poloprázdny, preusporiadaj symboly v blokoch $B_0, B_1, \ldots, B_{i-1}$ tak, aby boli uložené na spodných úrovniach blokov $B_1, B_2, \ldots, B_{i-1}$ 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_{-i} plný, preusporiadaj symboly na hornej úrovni bloku B_{-i} tak, aby boli uložené na spodných úrovniach blokov $B_{-(i-1)}, B_{-(i-2)}, \ldots, B_0$ a aby reprezentované slovo ostalo nezmenené.
7. Ak je blok B_{-i} poloprázdny, preusporiadaj symboly na spodnej úrovni bloku B_{-i} tak, aby boli uložené na spodných úrovniach blokov $B_{-(i-1)}, B_{-(i-2)}, \ldots, B_0$ 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 preusporiadavanie 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 poloformálnom 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. Ak navyše 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 Hennieho 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 preusporiadavanie blokov pomocou kopírovania na druhú pracovnú pásku stroja A, možno krok i-teho rádu zrejme odsimulovať v čase nanajvýš $c_1 \cdot 2^i$, 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, že medzi každými dvoma krokmi rádu väčšieho ako i je nutné vykonať aspoň jeden krok rádu i. Z toho možno pomocou jednoduchej 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 nanajvýš $T(n)/2^{i-2}$ 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. Hornou hranicou sumácie je $\log T(n) + c_2$, pretože 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 nanajvýš

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

kde c je vhodná konštanta. To dokazuje vetu 5.

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.