Univerzita Komenského, Bratislava
Fakulta matematiky, fyziky a informatiky

Výpočty konečných automatov s pomocnou informáciou

Diplomová práca

2016 **Bc. Rafael Korbaš**

Univerzita Komenského, Bratislava
Fakulta matematiky, fyziky a informatiky

Výpočty konečných automatov s pomocnou informáciou

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	2508 Informatika
Školiace pracovisko:	Katedra informatiky
Školiteľ:	prof. RNDr. Pavol Ďuriš CSc.

Bratislava, 2016

Bc. Rafael Korbaš

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Rafael Korbaš
Študijný program: informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Výpočty konečných automatov s pomocnou informáciou
Computations of finite automata with advice tape

Cieľ: Spracovaťprehľad o výsledkoch týkajúcich sa výpočtov konečných automatov s pomocnou informáciou a podobných výpočtových modelov. Porovnať výpočtovú silu deterministických a nedeterministických konečných automatov s pomocnou informáciou a skúmať výpočtovú zložitosť niektorých zaujímavých jazykov rozpoznávaných takýmito druhmi automatov.

Vedúci: prof. RNDr. Pavol Ďuriš, CSc.
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: doc. RNDr. Daniel Olejár, PhD.

Dátum zadania: 12.12.2014

Dátum schválenia: 17.12.2014

prof. RNDr. Branislav Rovan, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie

Touto cestou by som sa chcel poďakovať svojmu školiteľovi prof. RNDr. Pavlovi Ďurišovi CSc. za odbornú spoluprácu a trpezlivosť pri konzultáciách. Takisto ďakujem rodine a priateľom za podporu.

Abstrakt

Konečné automaty s pomocnou informáciou sú podobné klasickým konečným automatom s tým, že okrem vstupu disponujú navyše ďalším reťazcom, tzv. pomocnou informáciou. V našej práci táto pomocná informácia bude závisieť iba od dĺžky vstupu, t.j. pre všetky vstupy rovnakej dĺžky bude pomocná informácia rovnaká. Budeme skúmať predovšetkým vplyv spôsobu dodania pomocnej informácie, jej dĺžky vzhľadom na dĺžku vstupu a napokon aj determinizmu, resp. nedeterminizmu na výpočtovú silu modelu.

Kľúčové slová: konečný automat, pomocná informácia, výpočty s pomocnou informáciou, pomocná informácia zreťazená so vstupom, pomocná informácia na dodatočnej stope, pomocná informácia na dodatočnej páske

Abstract

Finite automata with advice are similar to standard finite automata, however apart from the input, they have access to an additional string, called an advice. In our work, this advice will only depend on the length of the input, i.e. for all the inputs of the same length the advice will be the same string. We will examine mainly the influence of the way of supplying the advice, it's length relative to the length of the input and finally the influence of the determinism and non-determinism on the power of the model

Key words: finite automata, advice string, advised computation, advice prefix, advice track, advice tape

Obsah

Úvod

Predpokladáme, že čitateľovi je v oblasti formálnych jazykov a automatov známy štandardný model deterministických a nedeterministických konečných automatov (skrátene DKA a NKA).

Konečné automaty s pomocnou informáciou sú podobné klasickým konečným automatom, no okrem vstupu disponujú navyše ďalším reťazcom, tzv. pomocnou informáciou. V našej práci táto pomocná informácia bude závisieť od dĺžky vstupu, t.j. pre všetky vstupy rovnakej dĺžky bude pomocná informácia rovnaká. Budeme skúmať predovšetkým vplyv spôsobu dodania pomocnej informácie, jej dĺžky vzhľadom na dĺžku vstupu a napokon aj determinizmu, resp. nedeterminizmu na výpočtovú silu modelu.

V minulosti boli skúmané predovšetkým modely konečných automatov s pomocnou informáciou zľava zreťazenou so vstupom (napr. [DH95]), pomocnou informáciou na dodatočnej stope (napr. [TYL03]) a napokon pomocnou informáciou na dodatočnej páske (napr. [KSY13]). V našej práci sú okrem iného zhrnuté výsledky z týchto článkov v kapitolách 2, 3 a 4.

Hlavným námetom pre náš výskum bol článok [KSY13], v ktorom sa autori venovali rôznym obmenám deterministických konečných automatov s pomocnou informáciou na dodatočnej páske. Výsledky z tohto článku sa nám podarilo rozšíriť napr. nájdením hierarchie jazykov s menšou ako lineárnou pomocnou informáciou (veta 4.3.6) a nájdením hornej hranice pre dĺžku pomocnej informácie, o ktorej má ešte zmysel uvažovať (veta 4.1.17), čo pokladáme za jeden z najväčších prínosov našej práce.

Ďalším prínosom tejto práce je výskum nedeterministických automatov s pomocnou informáciou na dodatočnej páske, ktorému sa dosiaľ, podľa našich informácii, nikto nevenoval. Viacero výsledkov bolo rovnakých s deterministickými automatmi, ale našlo sa takisto viacero rozdielov, napr. (ne)uzavretosť vzhľadom na komplement (dôsledok 4.2.6) a ďalšie, ktoré rozoberieme v nasledujúcich kapitolách.

Kapitola 1

Definície základných pojmov

Najskôr uvedieme základné definície a pojmy, na ktorých budeme stavať v neskorších kapitolách. V práci sa budeme držať notácie predovšetkým z [HMU01] a [KSY13].

1.1 Deterministický konečný automat

Definícia 1.1.1. ***Deterministický konečný automat (skrátene DKA)*** *A je pätica $(K, \Sigma, \delta, q_0, F)$, kde K je konečná množina stavov, Σ je konečná vstupná abeceda, $q_0 \in K$ je počiatočný stav, $F \subseteq K$ je množina (konečných) akceptačných stavov a $\delta : K \times \Sigma \to K$ je prechodová funkcia (δ-funkcia).*

Definícia 1.1.2. ***Konfigurácia*** *deterministického konečného automatu je prvok $(q, w) \in K \times \Sigma^*$, kde q je stav automatu a w je nespracovaná časť vstupného slova.*

Definícia 1.1.3. ***Krok výpočtu*** *deterministického konečného automatu A je relácia $\vdash_A$ na konfiguráciách definovaná $(q, av) \vdash_A (p, v) \iff p = \delta(q, a)$.*

Definícia 1.1.4. ***Jazyk*** *akceptovaný deterministickým konečným automatom A je množina $L(A) = \{w \mid w \in \Sigma^*, \exists q_F \in F, (q_0, w) \vdash_A^* (q_F, \epsilon)\}$*

1.2 Nedeterministický konečný automat

Definícia 1.2.1. ***Nedeterministický konečný automat*** *A je pätica $(K, \Sigma, \delta, q_0, F)$, kde K je konečná množina stavov, Σ je konečná vstupná abeceda, $q_0 \in K$ je počiatočný stav, $F \subseteq K$ je množina akceptačných (koncových) stavov a $\delta : K \times (\Sigma \cup \{\epsilon\}) \to 2^K$ je prechodová funkcia*

Definícia 1.2.2. ***Krok výpočtu*** *nedeterministického konečného automatu A je relácia $\vdash_A$ na konfiguráciách definovaná $(q, av) \vdash_A (p, v) \iff p \in \delta(q, a)$.*

Konfigurácia a jazyk akceptovaný NKA sú definované rovnako ako pre DKA. Narozdiel od DKA, minimálny NKA nemusí byť jednoznačný, teda môže existovať viacero možných minimálnych ekvivaletných NKA, ktoré nie sú navzájom izomorfné.

1.3 Pomocná informácia

Pojem pomocnej informácie sa dá definovať rôznymi spôsobmi. V našej práci a v článkoch, na ktorých staviame naše úvahy sa ***pomocná informácia*** chápe ako výstup ľubovoľnej funkcie $h : \mathbb{N} \to \Gamma^*$, pričom ako argument uvažujeme dĺžku vstupu pre konečný automat a jej hodnota je ľubovoľný reťazec nad abecedou Γ.

Pomocnú informáciu (p.i.) môžeme konečnému automatu dodať rôznymi spôsobmi:

1. p.i. zreťazíme so vstupom (podrobnejšie v **kapitole 2**).
2. p.i. dodáme na samostatnej stope vstupnej pásky (podrobnejšie v **kapitole 3**).
3. p.i. bude zapísaná na samostatnej páske (podrobnejšie v **kapitole 4**).

1.3.1 Rozdelenie jazykov podľa veľkosti potrebnej pomocnej informácie

Značná časť práce sa zaoberá zistením vplyvu veľkosti pomocnej informácie na výpočtovú silu zvolených modelov. Teraz zavedieme notáciu, ktorú budeme používať na rozlíšenie tried jazykov podľa veľkosti použitej pomocnej informácie pri ich akceptovaní.

Rozšírime notáciu z [KSY13]. Bude mať všeobecný tvar **MODEL**/$f(n)$ nasledované prípadne ďalšími upresneniami v zátvorke (definované v príslušných kapitolách). Za **MODEL** dosadíme príslušný model konečných automatov - definované v príslušných kapitolách. Ďalej $f(n)$ vyjadruje, že dĺžka pomocnej informácie je $O(f(n))$, kde n je dĺžka vstupu, prípadne **exp** je skratka pre pomocnú informáciu exponenciálnej dĺžky ($O(n2^n)$, ak sa inak nepovie, t.j pomocná informácia zodpovedajúca zoznamu všetkých binárnych reťazcov dĺžky n), **poly** pre pomocnú informáciu polynomiálnej dĺžky, t.j. $O(n^c)$, kde c je konštanta a **k**, resp. **const** - pomocná informácia konštantnej dĺžky.

Kapitola 2

Pomocná informácia pripojená pred vstup

V článku [DH95] autori uvažovali taký model deterministických konečných automatov s pomocnou informáciou, že pomocná informácia sa „pripojí" pred samotný vstup. Teraz uvedieme niektoré výsledky z tohto článku.

Na začiatok označme ako $\mathbf{REG}/f(n)$ triedu jazykov akceptovaných deterministickými konečnými automatmi, ktoré majú k dispozícií pomocnú informáciu dĺžky $f(n)$ pripojenú pred vstup, kde n vyjadruje dĺžku samotného vstupu.

Veta 2.0.1 ([DH95])**.** $\boldsymbol{REG}/k = \boldsymbol{REG}/poly$

Dôkaz. Nech jazyk L je akceptovaný nejakým automatom A s pomocnou informáciou polynomiálnej dĺžky $\alpha = (\alpha_n)$ (t.j. postupnosť reťazcov zo Σ^*, ktorých dĺžka závisí polynomiálne od dĺžky vstupu n). Inými slovami, automat A akceptuje slovo w, práve vtedy, keď sa po prečítaní reťazca $\alpha_{|w|}$ bude nachádzať v stave q, v ktorom keď začne čítať samotný vstup w, tak skončí v akceptačnom stave.

To znamená, že daný automat a pomocná informácia sa dajú nahradiť automatom A' a pomocnou informáciou β takými, že $\beta_{|w|}$ bude binárne kódovať počiatočný stav automatu A na slove w a zároveň A' bude mať oproti A upravenú „stromovú" prechodovú funkciu pomocou ktorej ekvivalentne spracuje pomocnú informáciu β. Zjavne pre daný jazyk stačí, aby pomocná informácia bola fixnej dĺžky, konkrétne horná hranica bude binárny logaritmus z počtu stavov automatu A', kde počet stavov je konštanta. □

Z toho vyplýva, že nemá zmysel zaoberať sa dlhšou ako konštantnou pomocnou informáciou. Zároveň ďalším postrehom je, že už jednobitová pomocná informácia stačí na to, aby výpočtová sila takýchto konečných automatov nebola porovnateľná s klasickými automatmi bez pomocnej informácie, či už konečnými, zásobnikovými, atď., nakoľko pre ľubovoľný unárny jazyk si pomocou jedného bitu môžeme zaznačiť, či

slovo do jazyka patrí alebo nie. Formálne $\mathbf{REG}/0 \subset \mathbf{REG}/1$. Ukazuje sa, že rôzna veľkosť konštantnej pomocnej informácie má vplyv na výpočtovú silu modelu.

V [DH95] na dokázanie tejto skutočnosti zaviedli pojem charakterizácie po častiach (*piecewise characterization*).

Veta 2.0.2 ([DH95])**.** *(Veta o charakterizácii po častiach) Nech $L \subseteq \Sigma^*$, potom nasledujúce tvrdenia sú ekvivalentné:*

1. $L \in \boldsymbol{REG}/k$,
2. *Existuje zobrazenie c:* $\mathbb{N} \to \{0, \ldots, 2^k - 1\}$ *a existujú regulárne jazyky* $A_0, \ldots, A_{2^k-1} \in \Sigma^*$ *také, že platí* $\forall n \in \mathbb{N} : A_{c(n)} \cap \Sigma^n = L \cap \Sigma^n$.

Dôkaz. $1 \implies 2$:
Preformulujme definíciu príslušnosti jazyka do $\mathbf{REG}/k$. Máme $L \in \mathbf{REG}/k$ práve vtedy, keď existuje regulárny jazyk A, zobrazenie $c : \mathbb{N} \to \{0, \ldots, 2^k - 1\}$ a slová $w_0, \ldots, w_{2^k-1} \in \{0,1\}^k$ také, že $\forall n \in \mathbb{N} : L \cap \Sigma^n = w_{c(n)}^{-1} A \cap \Sigma^n$, kde $w^{-1}A = \{u \in \Sigma^* | wu \in A\}$. Keďže pre ľubovoľný reťazec w je jazyk $w^{-1}A$ regulárny, keď aj A je regulárny, z toho priamo vyplýva tvrdenie 2.
$2 \implies 1$:
Nech A je zjednotenie jazykov $A_0, \ldots, A_{2^k-1}$ v tvare napr. $A = \bigcup_{i=1}^{2^k-1} bin(i)A_i$, kde $bin(i)$ je označenie pre binárne zakódované číslo i. Zrejme toto zjednotenie je tiež regulárne. Uvažujme teraz pomocnú informáciu konštantnej dĺžky $\alpha_n = bin(i)$, práve vtedy, keď $c(n) = i$. Zjavne jazyk L s pridanou pomocnou informáciou α zodpovedá regulárnemu jazyku A a teda L patrí do $\mathbf{REG}/k$. □

Ďalej sa v článku [DH95] autorom podarilo ustanoviť hierarchiu jazykov vzhľadom na rastúcu konštantnú veľkosť pomocnej informácie.

Veta 2.0.3 ([DH95])**.** $\forall k \geq 1 : \boldsymbol{REG}/(k-1) \subset \boldsymbol{REG}/k$.

Dôkaz. Indukciou. Dokážeme najprv $\mathbf{REG}/(1) \subset \mathbf{REG}/2$. Nech $\omega = \omega_0\omega_1 \ldots$ je náhodná nekonečná postupnosť núl a jednotiek a nech

$$L = \{bin(l)0^{n-1}i | \omega_{2n+l} = i, 0 \leq l < 2, n \geq 0\}.$$

Inými slovami, jazyk L obsahuje slová, v ktorých je zakódované, aký bit sa nachádza na $(2n + l)$-tej pozícii. Uvažujme regulárne jazyky:

$$L_{00} = \{0,1\}0^*0, L_{01} = 00^*0 \cup 10^*1, L_{10} = 00^*1 \cup 01^*0, L_{11} = \{0,1\}0^*1$$

Zrejme existuje zobrazenie $c : \mathbb{N} \to \{00, 01, 10, 11\}$ také, že pomocou neho a jazykov L_{00}, L_{01}, L_{10} a L_{11} po častiach poskladáme jazyk L, z čoho vyplýva, že $L \in \mathbf{REG}/2$.

Teraz ukážeme, že keby L patril do $\mathbf{REG}/1$, tak na reprezentáciu náhodného reťazca ω, resp. jeho prvých $2n$ bitov, by nám stačilo výrazne menej bitov ako je jeho dĺžka, čo je spor s náhodnosťou ω. Predpokladajme teda, že jazyk L by patril do $\mathbf{REG}/1$. Nech A je deterministický konečný automat, ktorý by akceptoval jazyk $\{\alpha_{|w|}w \mid w \in L\}$, kde $\alpha = (\alpha_n)$ je 1-bitová pomocná informácia. Ukážeme, že takto môžeme zostrojiť nasledovný program P, ktorý vypočíta prvých $2n$ bitov reťazca ω, pričom poznáme n a $\alpha_1, \ldots, \alpha_n$. Pseudokód:

read n;
for *i = 2 to n+1* **do**
  read α_i**;**
end
for *i = 2 to n+1* **do**
  if A **akceptuje** $\alpha_i 0^{i-1}0$ **then** $\omega_{2(i-2)} = 0$ **else** $\omega_{2(i-2)} = 1$**;**
  if A **akceptuje** $\alpha_i 0^{i-2}0$ **then** $\omega_{2(i-2)+1} = 0$ **else** $\omega_{2(i-2)+1} = 1$**;**
end

Program P môžeme implementovať na Turingovom stroji. Ako vstup dostane $n\#\alpha_2, \ldots, \alpha_{n+1}$. Na začiatku overí, či je vstup požadovaného tvaru, ak nie, zacyklí sa, ak áno, spustí na ňom program P, pričom bude simulovať konečný automat A. Na platných vstupoch takto vypočítame prvých $2n$ bitov reťazca ω, pričom nám stačilo na jeho reprezetáciu $O(log(n)) + n$ bitov (binárne zakódované n a $\alpha_2 \ldots \alpha_{n+1}$). To je v spore s náhodnosťou ω, nakoľko pre náhodný reťazec dĺžky $2n$ treba aspoň $2n - c$ bitov.

Túto myšlienku môžeme zovšeobecniť pre väčšie k nasledovne:

$$L = \{bin(l)0^{n-m}i | \omega_{kn+l} = i \text{ pre } 0 \leq l < k, n \geq 0\},$$

kde $m = \lfloor log(k) \rfloor$. Tým dostávame hierarchiu jazykov, na ktoré treba stále väčšiu a väčšiu pomocnú informáciu, t.j. L patrí do $\mathbf{REG}/k$, ale nepatrí do $\mathbf{REG}/(k-1)$.

□

2.0.2 Jazyk $\{a^n b^n \mid n \in \mathbb{N}\}$

Teraz sa pozrime na to, či konečné automaty s pomocnou informáciou zreťazenou so vstupom dokážu akceptovať jazyk $\{a^n b^n \mid n \in \mathbb{N}\}$. V [DH95] autori ukázali, že aj v prípade konečných automatov s pomocnou informáciou zreťazenou so vstupom sa dá aplikovať pumpovacia lema pre regulárne jazyky.

Veta 2.0.4 ([HMU01])**.** *(Pumpovacia lema pre regulárne jazyky) Ku každému regulárnemu jazyku L existuje číslo q také, že pre každé slovo $w \in L$ také, že $|w| > q$, existujú*

u, v, x také, že platí:

1. $w = uvx$
2. $|uv| \leq q$
3. $|v| \geq 1$
4. $\forall i \geq 0 : uv^i x \in L$

Autori v dôkaze takisto využili van der Waerdenovu vetu (2.0.5), ktorá hovorí, že nech akokoľvek ofarbíme všetky prirodzené čísla konečným počtom farieb, tak v tejto ofarbenej postupnosti existuje jednofarebná aritmetická postupnosť ľubovoľnej dĺžky. Formálne znenie tejto vety je nasledovné.

Veta 2.0.5 ([RG74]). *(van der Waerdenova veta) Pre ľubovoľné, $r, k \in \mathbb{N}, r, k > 0$ existuje číslo n také, že ak ofarbíme čísla $\{1, 2, \ldots, n\}$ r rôznymi farbami, potom existuje aspoň k čísel tej istej farby takých, že tvoria aritmetickú postupnosť.*

Dôkaz. Dôkaz možno nájsť v článku [RG74]. □

Priamym dôsledkom vety 2.0.5 je lema 2.0.6, ktorú použijeme pri dôkaze vety 2.0.7.

Lema 2.0.6 ([DH95]). *Nech $N_{\geq q} = \{n \mid n \geq q\}$ pre ľubovoľné $q \in \mathbb{N}$ a $S_{a,b} = \{an + b \mid n \in \mathbb{N}\}$ pre ľubovoľné $a, b \in \mathbb{N}$. Nech c je r-farbenie množiny prirodzených čísel, t.j. $c : \mathbb{N} \to \{0, 1, \ldots, r-1\}$ a nech l je ľubovoľná konštanta. Potom existuje jednofarebná aritmetická postupnosť v $N_{\geq q} \cap S_{a,b}$ dĺžky l.*

Veta 2.0.7 ([DH95]). $\{a^n b^n \mid n \in N\} \notin \boldsymbol{REG}/const.$

Dôkaz. Predpokladajme, že by platilo $L = \{a^n b^n \mid n \in \mathbb{N}\} \in \mathbf{REG}/k$ pre nejaké $k \in \mathbb{N}$. Potom na základe vety 2.0.2 existuje zobrazenie $c : \mathbb{N} \to \{0, \ldots, 2^k - 1\}$ a regulárne jazyky $A_0, \ldots, A_{2^k-1}$ také, že pre všetky $n \in \mathbb{N}$ platí $L \cap \Sigma^n = A_{c(n)} \cap \Sigma^n$.

Nech $q = max\{q_i \mid 0 \leq i \leq 2^k - 1\}$, kde q_i je konštanta z pumpovacej lemy pre regulárne jazyky (2.0.4) pre jazyk A_i. Zobrazenie c zodpovedá 2^k-farbeniu prirodzených čísel. S použitím lemy 2.0.6, ak zvolíme $l = q+1$ a množinu $S_{2,0}$ dostávame, že existuje jednofarebná aritmetická postupnosť tvaru:

$$P_1 = \{2n, 2n + d, \ldots, 2n + (l-1) \cdot d\} \subseteq N_{\geq 2q} \cap S_{2,0}.$$

Bez ujmy na všeobecnosti môžeme predpokladať, že čísla z postupnosti P_1 majú farbením c priradenú farbu 0. V tom prípade slovo $a^n b^n$ patrí do jazyka A_0.

Keďže $n > q$, môžeme slovo $a^n b^n$ rozdeliť podľa pumpovacej lemy pre regulárne jazyky na úseky u, v, x také, že:

1. $|uv| \leq q$,

2. $|v| \neq 0$,

3. $\forall i \in \mathbb{N} : uv^i x \in A_0$.

Dĺžky slov tvaru $uv^i x$ tvoria aritmetickú postupnosť

$$P_2 = \{2n, 2n + |v|, 2n + 2|v|, \dots\} \subseteq N_{\geq 2q}.$$

Môžeme si všimnúť, že $2n + |v| \cdot d$ patrí do P_1 a zároveň do P_2, z čoho vyplýva, že $uv^{d+1}x$ patrí do A_0 a teda aj do L. Zároveň ale slovo $uv^{d+1}x$ nemôže mať rovnaký počet znakov a a b, keďže slovo uv je dĺžky $q < n$ a teda tvaru a^*. Teda $uv^{d+1}x \notin L$, čo je spor, z čoho vyplýva, že jazyk $\{a^n b^n | n \in \mathbb{N}\} \notin \mathbf{REG}/const$.

□

Kapitola 3

Pomocná informácia zapísaná na dodatočnej stope

V článku [TYL03] sa autori zamerali na model deterministických konečných automatov s pomocnou informáciou zapísanou na dodatočnej stope vstupnej pásky, čo umožňuje hlave čítať súčasne pomocnú informáciu a vstup. Triedu jazykov akceptovaných takýmito automatmi budeme označovať **REG**/n (*advice track*). Formálna definícia:

Definícia 3.0.8. *Jazyk $L \subseteq \Sigma^* \in$ **REG**/n (advice track), ak existuje deterministický konečný automat A a funkcia pomocnej informácie $h : \mathbb{N} \to \Sigma^*$ taká, že $|h(n)| = n$, pričom $\forall x \in \Sigma^* : x \in L \iff [^{x}_{h(|x|)}] \in L(A)$, kde $[^{x}_{h(|x|)}]$ vyjadruje, že na vstupnej páske automatu A je na vrchnej stope zapísaný reťazec x a na spodnej, teda pomocnej stope vstupnej pásky, je zapísaný reťazec $h(|x|)$.*

Ľahko sa dá nahliadnuť, že s takýmto druhom pomocnej informácie je možné akceptovať napríklad bezkontextový jazyk $\{a^n b^n | n \in \mathbb{N}\}$, o ktorom bolo ukázané v [DH95], že sa nedá akceptovať na konečných automatoch, ktoré majú pomocnú informáciu zreťazenenú so vstupom (veta 2.0.7). Ale ako teraz ukážeme, bezkontextové jazyky (značíme CFL) nie sú podmnožinou triedy jazykov **REG**/n (*advice track*).

3.1 Porovnanie s triedou bezkontextových jazykov

Veta 3.1.1 ([TYL03])**.** *Majme jazyk $EQUAL_{0,1} = \{x \in \{0,1\}^* | \#_0(x) = \#_1(x)\}$. Platí $EQUAL_{0,1} \notin$ **REG**/n (advice track), teda $CFL \not\subseteq$ **REG**/n (advice track).*

Dôkaz. Nech $\Sigma = \{0,1\}$. Predpokladajme, že by platilo $EQUAL_{0,1} \in$ **REG**/n (*advice track*). Potom existuje deterministický konečný automat $A = (Q, \Sigma, q_0, F)$ a funkcia pomocnej informácie h také, že $\forall x \in \Sigma^*$, $x \in EQUAL_{0,1} \iff [^{x}_{h(|x|)}] \in L(A)$.

Nech n označuje počet stavov automatu A, t.j. $n = |Q|$. Pre každé celé číslo $k \in \{0, \ldots, n\}$ bude y_k vyjadrovať ľubovoľný reťazec dĺžky n nad abecedou $\{0, 1\}$ taký, že sa v ňom znak 0 nachádza práve k-krát, t.j. $\#_0(y_k) = k$.

Z Dirichletovho princípu ($n+1 > |Q|$) vyplýva, že existuje dvojica rôznych indexov $k, l \in \{0, \ldots, n\}$ a reťazce $z_k, z_l \in \Sigma^*$ také, že platia súčasne nasledujúce podmienky:

1. $y_k z_k, y_l z_l \in EQUAL_{0,1}$,
2. automat A sa po prečítaní reťazca $\left[\begin{smallmatrix} y_k \\ w_n \end{smallmatrix}\right]$ nachádza v tom istom stave ako po prečítaní $\left[\begin{smallmatrix} y_l \\ w_n \end{smallmatrix}\right]$, kde w_n je prvých n znakov $h(2n)$.

Z toho vyplýva, že A akceptuje aj reťazec $\left[\begin{smallmatrix} y_k z_l \\ h(y_k z_l) \end{smallmatrix}\right]$, pričom by malo platiť, že $\#_0(y_k z_l) = \#_1(y_k z_l)$, a teda $\#_0(z_l) = n - k$. To by ale znamenalo, že $\#_0(y_l) = k$, keďže $\#_0(y_l z_l) = \#_1(y_l z_l)$. To je ale spor s definíciou y_l, z čoho vyplýva, že $EQUAL_{0,1} \notin REG/n$, a teda $CFL \not\subseteq \mathbf{REG}/n$ (*advice track*). □

3.2 Porovnanie s Turingovými strojmi

Je známe, že trieda regulárnych jazykov (REG) zodpovedá triede jazykov akceptovaných na jednopáskových deterministických Turingových strojoch fungujúcich v $o(n\,log(n))$ čase, kde n je dĺžka vstupu, pričom toto ohraničenie je optimálne, lebo v čase $O(n\,log(n))$ sa na Turingovom stroji dá akceptovať napríklad jazyk $\{a^n b^n | n \in \mathbb{N}\}$, ktorý nie je regulárny ([TYL03]).

Nech 1-$DLIN/lin$ označuje triedu všetkých jazykov L takých, že existuje jednopáskový deterministický Turingov stroj počítajúci v čase $O(|w|)$ (kde w je vstup, a teda $|w|$ jeho dĺžka), funkcia pomocnej informácie h a konštanta $c \geq 1$ také, že:

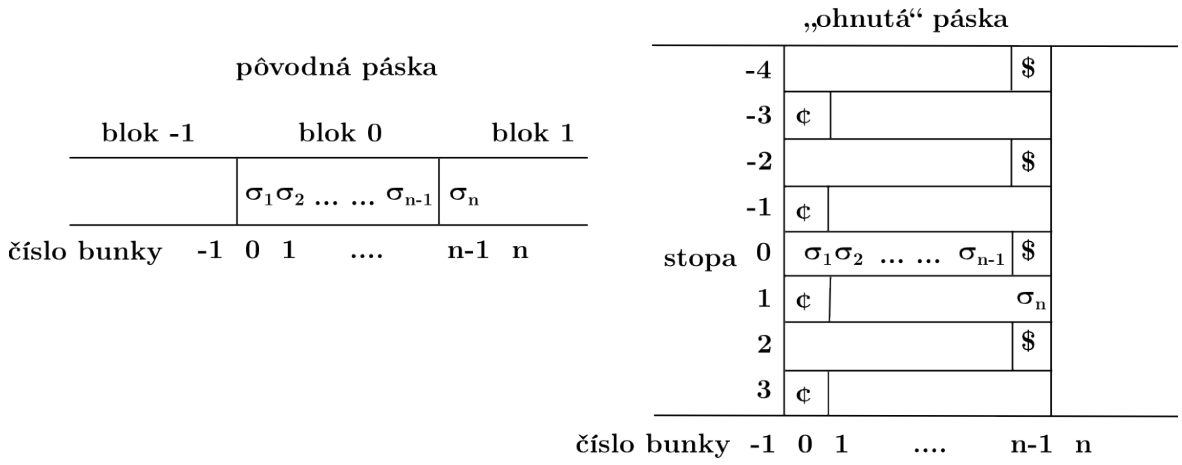
1. $\forall n \in \mathbb{N} : |h(n)| \leq cn + c$,
2. $\forall x : x \in L \iff \left[\begin{smallmatrix} x \\ h(|x|) \end{smallmatrix}\right]$.

O triede $\mathbf{REG}/n$ (*advice track*) sa autorom v článku [TYL03] podarilo dokázať nasledujúcu vetu:

Veta 3.2.1 ([TYL03])**.** ***REG***$/n$ (*advice tape*) = 1-$DLIN/lin$

Skôr ako dokážeme samotnú vetu 3.2.1, definujme konštrukciu Turingovho stroja „ohýbajúceho“ pásku tak, aby mala vyhovujúcu dĺžku, pričom čas výpočtu na takto upravenej páske sa nezmení. Túto konštrukciu potom využijeme v samotnom dôkaze.

Hlavná myšlienka konštrukcie je, že pridáme na vstupnú pásku dostatočný (ale stále konštantný) počet prázdnych stôp (simulujúcich blanky na pôvodnej páske) tak,



Obr. 3.1: Príklad úpravy pásky na stroji N pre $k = 2$. Pôvodná páska M je rozdelená na $4k$ blokov veľkosti $n - 1$ a každý blok je na N simulovaný na samostatnej stope. Napríklad blok 0 je simulovaný stopou s indexom 0, blok 1 je zasa simulovaný stopou s indexom 1, ale v opačnom smere.

aby Turingov stroj pri výpočte nemusel opustiť oblasť pásky, kde je zapísaný vstupný reťazec.

Konštrukcia ohýbajúceho Turingovho stroja ([TYL03]).
Nech $M = (Q, \Sigma, \Gamma, \delta, q_0', q_{acc}, q_{rej})$ je ľubovoľný jednopáskový Turingov stroj, ktorý zastane v lineárnom čase.

Ohýbajúci Turingov stroj N zostrojíme z M nasledovne.

Vyberme najmenšie kladné celé číslo k také, že $k|x|$ zhora ohraničuje dĺžku výpočtu pre všetky reťazce x dlhé aspoň 3, t.j. formálne $\forall x \in \Sigma*, |x| \geq 3 : Time_M(x) \leq k|x|$.

Všimnime si, že hlava M sa môže pohybovať po páske v oboch smeroch, teda môže využiť priestor naľavo aj napravo od vstupného slova, ale tento priestor je vzhľadom na hornú hranicu času výpočtu ohraničený. Ak prvý znak vstupného slova x má index 0, môžeme pohyb hlavy na páske ohraničiť zľava indexom $-2k(|x| - 1)$ a sprava $2k(|x| + 1) - 1$.

Ďalej do automatu N pridajme štyri stavy $q_0, q_1, q_2, q_3 \notin Q$ a stavy tvaru $\left[\begin{smallmatrix} i \\ q \end{smallmatrix}\right]$ pre všetky $i \in \{-2k, -2k+1, \ldots 2k-1\}$ a všetky stavy $q \in Q$. Nech x je ľubovoľný reťazec zapísaný na vstupnej páske stroja M. Výpočet stroja N bude prebiehať nasledovne.

1. Stroj N začne v stave q_0. Ak je vstup prázdny, tak N hneď vstúpi do koncového stavu M, bez toho, aby hýbal hlavou. Predpokladajme ďalej, že vstup je neprázdny reťazec $x = \sigma_1\sigma_2 \ldots \sigma_n$, kde σ_i sú znaky abecedy Σ.

2. Následne stroj N predspracuje vstup tak, ako vidno na obrázku 3.1. Políčka na páske s indexmi $-2k(|x| - 1)$ až $2k(|x| + 1) - 1$ rozdelíme na $4k$ blokov dĺžky $|x| - 1$. Bloky indexujeme zľava doprava $-2k$ až $2k - 1$. Pracovnú pásku stroja N

potom rozdelíme na $4k$ stôp. Intuitívne, chceme simulovať na i-tej stope ohnutej pásky stroja N blok s indexom i na pracovnej páske stroja M.

Stroj N na začiatku umiestni na všetky stopy s nepárnym indexom na políčku s indexom 0 znak ¢(označenie ľavého konca). Následne stroj prejde do stavu q_1 a hlava sa bude hýbať doprava, dokým neuvidí prvý blank. Ak $|x| \geq 3$, tak N vstúpi do stavu q_2 a hlava urobí krok späť, inak N vstúpi do koncového stavu stroja M.

Ďalej N umiestni na danom políčku na všetky párne stopy znak \$ (označenie pravého konca) a posunie znak σ_n zo stopy 0 na stopu 1, vstúpi do stavu q_3 a hlava sa vráti na počiatočné políčko pásky.

3. Stroj N bude simulovať na takto upravenej páske pohyb stroja M po pôvodnej pracovnej páske. Keď sa M bude nachádzať v bloku i v stave q, tak N sa bude nachádzať v stave $\begin{bmatrix} i \\ q \end{bmatrix}$. Ak je i párne, tak N bude hýbať hlavu v tom istom smere ako M, inak pôjde v opačnom smere.

   Ešte treba vyriešiť prípad, keď hlava M prejde z jedného bloku do druhého. Ak má blok párny index, nech je to $2j$, tak v prípade, že M tento blok opúšťa sprava a prejde do stavu q, tak n prejde do stavu $\begin{bmatrix} 2j+1 \\ q \end{bmatrix}$ a hlava sa posunie na páske doľava, ak zľava, tak analogicky prejde do stavu $\begin{bmatrix} 2j-1 \\ q \end{bmatrix}$ a hlava sa posunie doprava. Ak M pred posunom zapísal nejaký znak na pásku, nech je to σ, tak N ho zapíše na príslušné miesto na $2j$-tej stope. V prípade opustenia nepárneho bloku je to analogické. Zrejme N počas simulácie stroja M nikdy neopustí oblasť pásky, kde bol vstup a výpočet N má rovnako veľa krokov ako výpočet M.

Dôkaz vety 3.2.1. Inklúzia $\mathbf{REG}/n$ (*advice track*) $\subseteq$ 1-$DLIN/lin$ je zrejmá. Teraz ukážeme opačnú inklúziu, teda, že 1-$DLIN/lin \subseteq \mathbf{REG}/n$ (*advice track*).

Nech L je ľubovoľný jazyk nad abecedou Σ patriaci do triedy 1-$DLIN/lin$. Bez ujmy na všeobecnosti, vezmime jednopáskový deterministický Turingov stroj M a funkciu pomocnej informácie h, pre ktorú platí:

1. $\forall n \in \mathbb{N} : n \leq |h(n)| \leq cn$,
2. $\forall x : x \in L \iff \left[{}^{x}_{h(|x|)}\right] \in L(M)$.

Pre jednoduchosť uvažujme, že abeceda pomocnej informácie je odlišná od Σ.

Nech x je ľubovoľný vstup dĺžky n. Na začiatku sa na páske Turingovho stroja M nachádza reťazec $\begin{bmatrix} x\#^{|h(n)|-n} \\ h(n) \end{bmatrix}$. Tento vstup môžeme prerobiť na vstup lineárnej dĺžky tak, že obsah pásky „poskladáme“ na dĺžku n podobne ako v konštrukcii ohýbajúceho Turingovho stroja s tým rozdielom, že stopy nebudú obsahovať falošné blanky, ale

mriežky, resp. zodpovedajúcu časť $h(n)$. Túto operáciu nevykonávame na stroji, ide len o to vyjadriť iným spôsobom pomocnú informáciu, takže na časovú zložitosť samotného výpočtu toto „poskladanie" nemá vplyv. Označme takto upravený vstup $cont(x, h(n))$.

Teraz spustime na $cont(x, h(n))$ ohýbajúci Turingov stroj N, ktorého konštrukciu sme vyššie opísali. Z $cont(x, h(n))$ môžeme vymazať vstup x, čiže tam ostane len poskladaná pomocná informácia a padding v podobe mriežok. Tento reťazec môžeme použiť ako novú pomocnú informáciu, nazvime ju $h'(n)$. Všimnime si, že $|h'(n)| = n$.

Definujme teraz nový jednopáskový Turingov stroj M' taký, že na vstupe dostane reťazec $\left[{}^{2j+1}_{q}\right]$. Tento vstup v lineárnom čase prerobí na $cont(x, h(n))$ (stačí mu len na správne miesto „vsunúť" vstup x) a na takto upravenom vstupe odsimuluje stroj N. Zrejme pre každý vstup x platí, že $x \in L \iff M'$ akceptuje $[{}^{x}_{h'(|x|)}]$, pričom M' beží v lineárnom čase pohybujúc sa hlavou po páske len vrámci vstupného reťazca. Takýto stroj môžeme na základe skutočnosti, že jednopáskové Turingove stroje pracujúce v $o(log(n))$ čase akceptujú práve regulárne jazyky ([Kob85]), previesť na ekvivalentný konečný automat. Z toho vyplýva, že $L \in \mathbf{REG}/n(advice\ track)$.

□

3.3 Zamieňacia lema a jazyk palindrómov

V tejto časti sa pozrieme na jazyk $PAL = \{ww^R | w \in \{0,1\}^*\}$. V článku [Yam08] autor ukázal, že tento jazyk sa nedá akceptovať na deterministických konečných automatoch s pomocnou informáciou na dodatočnej stope. Pri dôkaze využil tzv. *zamieňaciu lemu (anglicky Swapping lemma)* pre regulárne jazyky, ktorú teraz uvedieme a dokážeme. Pred dôkazom ešte poznamenávame, že na označenie prefixu, resp. sufixu slova w po, resp. od i-teho znaku budeme používať značenie $pref_i(w)$, resp. $suf_i(w)$.

Lema 3.3.1 ([Yam08])**.** *(Zamieňacia lema pre regulárne jazyky) Nech L je ľubovoľný nekonečný regulárny jazyk nad abecedou Σ, $|\Sigma| \geq 2$. Potom existuje kladné celé číslo m (tzv. konštanta zamieňacej lemy) také, že pre ľubovoľné celé čislo $n \geq 1$ a ľubovoľnú podmnožinu S množiny $L \cap \Sigma^n$ s aspoň m prvkami platí, že pre ľubovoľné $i \in \{0, \ldots, n\}$ existujú reťazce $x = x_1x_2$, $y = y_1y_2$ v S také, že $|x_1| = |y_1| = i$, $|x_2| = |y_2|$ a zároveň platí:*

1. $x \neq y$,
2. $y_1x_2 \in L$,
3. $x_1y_2 \in L$.

Dôkaz. Nech L je ľubovoľný regulárny jazyk nad abecedou Σ. Nech $M = (Q, \Sigma, \delta, q_0, F)$ je deterministický konečný automat akceptujúci daný jazyk, kde Q je množina stavov automatu, δ je jeho prechodová funkcia, $q_0 \in Q$ počiatočný stav a $F \subseteq Q$ množina konečných stavov.

Nech naša konštanta zamieňacej lemy, $m = |Q|$, nech n je aspoň 1 a S nech je ľubovoľná podmnožina množiny $L \cap \Sigma^n$, pričom $|S| > m$. Zrejme $|S| \geq 2$, keďže M má aspoň jeden stav.

Zvoľme si teraz ľubovoľný index $i \in \{0, \ldots, n\}$. Prípady $i = 0$ a $i = n$ sú triviálne, keďže v tom prípade stačí zvoliť ľubovoľné dva reťazce $x, y \in S$.

Nech teda $2 \leq i \leq n-1$. Uvažujme možné vnútorné stavy automatu M pri prečítaní i-teho znaku vstupu. Keďže $|S| > |Q|$, existujú dva odlišné reťazce $x, y \in S$, pre ktoré M bude po prečítaní i-teho znaku reťazca x, resp. y v tom istom stave, označme ho q. Označme prefix reťazca x po i-ty znak x_1 a sufix x_2, pre reťazec y analogicky. Keďže M sa nachádza po prečítaní x_1 a y_1 v tom istom stave q, teda nevie ich od seba rozlíšiť a $x = x_1x_2$ aj $y = y_1y_2$ patria do jazyka L, tak M musí akceptovať aj reťazce x_1y_2, resp. y_1x_2. □

Túto lemu teraz môžeme aplikovať na jazyk palindrómov párnej dĺžky.

Veta 3.3.2 ([Yam08])**.** $PAL = \{ww^R | w \in \{0,1\}^*\} \notin \boldsymbol{REG}/n$ (*advicetrack*).

Dôkaz. Predpokladajme, že by jazyk PAL bol v $\mathbf{REG}/n$ (*advicetrack*). Potom existuje regulárny jazyk L nad abecedou Σ a funkcia pomocnej informácie h také, že pre každý reťazec $x \in \{0,1\}^*$ platí $[{}^{x}_{h(|x|)}] \in L \iff x \in PAL$.

Uvažujme konštantu m zo zamieňacej lemy pre jazyk L. Nech $n = 2m$ a $i = \frac{n}{2}$. Nech podmnožina S množiny $L \cap \Sigma^n$ je $S = \{[{}^{x}_{h(n)}] \,||x| = n\}$.

Všimnime si, že $|S| \geq 2^{\frac{n}{2}} > m$. Podľa zamieňacej lemy existujú dva rôzne reťazce $x, y \in \{0,1\}^n$ také, že $[{}^{x}_{h(n)}]$ aj $\left[{}^{y}_{h(n)}\right]$ patria do S. Nech $u_1 = pref_i(x)$ a $u_2 = pref_i(y)$. Potom môžeme reťazce x a y vyjadriť ako $x = u_1u_1^R$, $y = u_2u_2^R$.

Uvažujme teraz reťazce $u_1u_2^R = pref_{\frac{n}{2}}(x)suf_{\frac{n}{2}}(y)$ a $u_2u_1^R = pref_{\frac{n}{2}}(y)suf_{\frac{n}{2}}(x)$. Tieto reťazce zjavne nepatria do jazyka PAL, lebo nie sú tvaru ww^R, a teda reťazce $\left[{}^{u_1u_2^R}_{h(n)}\right]$, resp. $\left[{}^{u_2u_1^R}_{h(n)}\right]$ nepatria do jazyka L, čo je v spore s tým, že jazyk L je regulárny, a teda nemôže byť akceptovaný na žiadnom konečnom automate, z čoho vyplýva $PAL \notin \mathbf{REG}/n$ (*advice track*). □

3.4 Zovšeobecnená zamieňacia lema a vzťah k deterministickým bezkontextovým jazykom

Teraz uvedieme zovšeobecnenú verziu zamieňacej lemy. V leme 3.3.1 reťazce x a y sú oba rozdelené na dva bloky a jeden z blokov je použitý pri zámene. V nasledujúcej leme však reťazce rozdelíme na ľubovoľný fixný počet blokov, pričom jeden z nich je použitý pri zámene. Táto forma zamieňacej lemy sa nám zvlášť zíde pri dôkaze, že deterministický bezkontextový jazyk $GT = \{w | w \in \{0,1\}^* \#_0(w) > \#_1(w)\}$ nepatrí do $\mathbf{REG}/n(advice\ track)$.

Lema 3.4.1 ([Yam08]). *(Zovšeobecnená zamieňacia lema pre regulárne jazyky) Nech L je ľubovoľný nekonečný regulárny jazyk nad abecedou Σ, pričom $|\Sigma| \geq 2$. Potom existuje celé kladné číslo m také, že pre každé $n \geq 1$, ľubovoľnú množinu $S \subseteq L \cap \Sigma^n$ a ľubovoľnú postupnosť $(i_1, i_2, \ldots, i_k) \in \{1, \ldots, n\}^k$, kde $\sum_{j=1}^{k} i_j = n$, pre niektoré $k \in \{1, \ldots, n\}$ platí preň nasledujúce tvrdenie:*

Ak $|S| > m$, tak existujú dva reťazce $x = x_1 x_2 \ldots x_{k+1}$ a $y = y_1 y_2 \ldots y_{k+1}$ v množine S také, že $|x_{k+1}| = |y_{k+1}|$ a $|x_{j'}| = |y_{j'}| = i_{j'}$ pre každý index $j' \in \{1, \ldots, k\}$, taký, že pre každý index $j \in \{1, \ldots, k\}$ máme:

1. *$x \neq y$,*
2. *$x_1 \ldots x_{j-1} y_j x_{j+1} \ldots x_{k+1} \in L$,*
3. *$y_1 \ldots y_{j-1} x_j y_{j+1} \ldots y_{k+1} \in L$.*

Dôkaz. Všimnime, si, že prípad $k = 1$ je pokrytý Lemou 3.3.1. Uvažujme deterministický konečný automat $M = (Q, \Sigma, \delta, q_0, F)$ akceptujúci jazyk L. Nech pre $S \subseteq L \cap \Sigma^n$ platí, že $|S| > m$, kde $m = |Q|^k$.

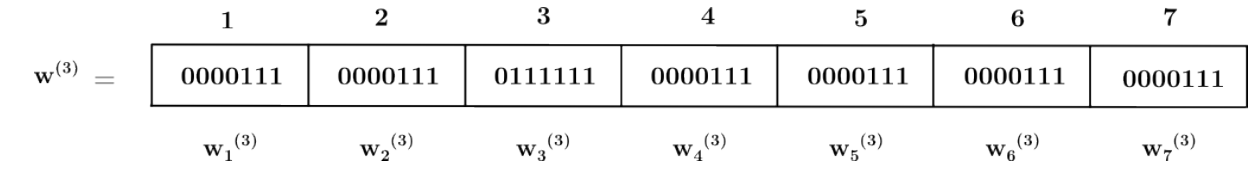
Každému reťazcu $s \in S$ priradíme k-ticu $(q_1, \ldots, q_k)$ pozostávajúcu z vnútorných stavov automatu M takých, že pre každé $j \in \{1, \ldots, k\}$ M vstúpi do stavu q_j po prečítaní $(\sum_{e=1}^{j} i_e)$-teho znaku daného reťazca. Takýchto k-tic je nanajvýš $|Q_k|$.

Keďže $|S| > |Q|^k$, tak existujú dva rôzne reťazce $x, y \in S$ také, že majú priradenú tú istú postupnosť vnútorných stavov, nech je to $(q_1, q_2, \ldots, q_k)$. Rozpíšme si x a y ako $x = x_1 x_2 \ldots x_{k+1}$ a $y = y_1 y_2 \ldots y_{k+1}$, kde $|x_{k+1}| = |y_{k+1}|$ a $|x_{j'}| = |y_{j'}| = i_{j'}$ pre každý index $j' \in \{1, \ldots, k\}$. Všimnime si, že pre každý index $j \in \{1, \ldots, k\}$ M vstúpi do toho istého vnútorného stavu q_j po prečítaní bloku x_j v reťazci x, resp. bloku y_j v reťazci y.

Zafixujme si ľubovoľný index $j \in \{1, \ldots, k\}$. Na základe spôsobu, akým sme zvolili x a y, môžeme zameniť bloky x_j a y_j v x a y bez toho, aby sme ovplyvnili akceptáciu vstupu automatom M.

Z toho vyplýva, že upravené reťazce $x_1 \dots x_{j-1} y_j x_{j+1} \dots x_{k+1}$ a $y_1 \dots y_{j-1} x_j y_{j+1} \dots y_{k+1}$ sú oba akceptované automatom M, čo je to, čo sme chceli dokázať. □

Túto lemu teraz využijeme pri dôkaze tvrdenia, že jazyk GT sa nedá akceptovať na deterministických konečných automatoch s pomocnou informáciou na dodatočnej stope.



Obr. 3.2: Reťazec $w^{(j)}$ pre $j = 3$ a $m = 7$

Veta 3.4.2 ([Yam08])**.** $GT = \{w | w \in \{0,1\}^* \#_0(w) > \#_1(w)\} \notin \boldsymbol{REG}/n(\textit{advice track})$.

Dôkaz. Predpokladajme, že by jazyk GT patril do $\mathbf{REG}/n(\textit{advice track})$. Potom existuje funkcia pomocnej informácie h a regulárny jazyk L nad abecedou Σ také, že pre každý binárny reťazec x, $x \in GT \iff [{}^{x}_{h(|x|)}] \in L$. Nakoľko L je nekonečný regulárny jazyk, môžeme použiť Lemu 3.4.1.

Nech m je konštanta zovšeobecnenej zamieňacej lemy. Bez ujmy na všeobecnosti môžeme predpokladať, že m je nepárne a aspoň 3. Nech $n = m^2$. Zamerajme sa na množinu $L \cap \Sigma^n$. Nech $(i_1, i_2, \dots, i_{m-1})$ je postupnosť čísel jednoznačne určená tým, že $i_j = m$ pre každý index $j \in \{1, \dots m-1\}$. Táto postupnosť čísel rozdeľuje každý n-bitový reťazec na m blokov rovnakej veľkosti m.

Pre každý index $j \in \{1, \dots, m-1\}$, nech $w^{(j)}$ označuje reťazec $w_1^{(j)} w_2^{(j)} \dots w_m^{(j)}$ s nasledujúcimi vlastnosťami:

1. $w_j^{(j)} = 01^{m-1}$,
2. $i \neq j : w_i^{(j)} = 0^{m'+1} 1^{m'}$, kde $m' = \lfloor \frac{m}{2} \rfloor$.

Na obrázku 3.2 môžeme vidieť, ako vyzerá napríklad $w^{(j)}$ pre $j = 3$ a $m = 7$. Keďže $\#_0(w^{(j)}) = \#_1(w^{(j)}) + 1$, tak reťazec $w^{(j)}$ patrí do jazyka GT. Množina S je teda definovaná ako $\{ \left[{}^{w^{(1)}}_{h(n)}\right], \dots, \left[{}^{w^{(m)}}_{h(n)}\right] \}$. Zjavne $|S| \geq m$. Podľa lemy 3.4.1 existujú dva odlišné reťazce $w^{(k)}$ a $w^{(l)}$, pričom $k \neq l$, také, že $\left[{}^{w^{(k)}}_{h(n)}\right], \left[{}^{w^{(l)}}_{h(n)}\right] \in S$ a $\left[{}^{\widetilde{w}^{(k)}}_{h(n)}\right], \left[{}^{\widetilde{w}^{(l)}}_{h(n)}\right] \in L$, kde $\widetilde{w}^{(k)}$ a $\widetilde{w}^{(l)}$ dostaneme z $w^{(k)}$, resp. $w^{(l)}$ tým, že zameníme navzájom ich l-té bloky.

Všimnime si, že platí:

$$\forall i \in \{0,1\} : \#_i(\widetilde{w}^{(k)}) = \#_i(w^{(k)}) - \#_i(w_l^{(k)}) + \#_i(w_l^{(l)}).$$

Z toho, že $\#_0(w_l^{(k)}) = \#_1(w_l^{(k)}) + 1$, $\#_0(w_l^{(l)}) = 1$ a $\#_1(w_l^{(l)}) = m - 1$ priamo vyplýva, že $\#_1(\widetilde{w}^{(k)}) = \#_0(\widetilde{w}^{(k)}) + m - 2$, čo je väčšie ako $\#_0(\widetilde{w}^{(k)})$, keďže m je podľa predpokladov aspoň 3. Z toho vyplýva, že $\widetilde{w}^{(k)} \notin GT$, čo je v spore s lemou 3.4.1, takže jazyk GT nemôže patriť do $\mathbf{REG}/n(advice\ track)$. □

Kapitola 4

Konečné automaty s pomocnou páskou

V tejto kapitole sa zameriame na konečné automaty s pomocnou informáciou na dodatočnej páske. V článku [KSY13] autori skúmali predovšetkým deterministické konečné automaty s pomocnou páskou (skrátene $DFAT$). V našej práci k výsledkom z článku [KSY13] pridávame ďalšie výsledky a analyzujeme nedeterministickú verziu týchto automatov (skrátene $NFAT$).

4.1 Deterministické konečné automaty s pomocnou páskou

4.1.1 Definície

V tejto časti budeme definovať deterministický konečný automat s pomocnou páskou (skrátene $DFAT$). Nakoľko budeme uvažovať viacero variantov a budeme v niektorých prípadoch pripúšťať pohyb hlavy, resp. hláv rôznymi smermi, budeme niektoré veci definovať všeobecnejšie, ako keby sme vychádzali priamo z definície deterministických konečných automatov. Na $DFAT$ sa môžeme pozerať ako na jednosmerný (prípadne viacsmerný) Turingov stroj s dvomi, prípadne viacerými páskami bez možnosti zapisovať, pričom na prvej páske je zapísaný vstup a na druhej, resp. ostatných páskach je zapísaná pomocná informácia. Pomocná informácia je určená funkciou, ktorá ako vstup dostane celé číslo, konkrétne dĺžku vstupu a jej výstup je reťazec znakov pracovnej abecedy. Následne automat číta obsahy týchto pások a akceptuje, resp. neakceptuje vstup podobne ako štandardný konečný automat.

Definícia 4.1.1. ***Deterministický konečný automat s pomocnou páskou (označujeme*** $DFAT$***)*** *A je sedmica* $(K, \Sigma, \Gamma, \delta, q_0, F, h)$*, kde* K *je konečná množina stavov,*

Σ *je konečná vstupná abeceda,* Γ *je konečná abeceda pomocnej informácie,* $q_0 \in K$ *je počiatočný stav,* $F \subseteq K$ *je množina akceptačných stavov,* $\delta : K \times (\Sigma \cup \{¢, \$\}) \times (\Gamma \cup \{¢, \$\}) \to K \times \{0,1\} \times \{0,1\}$ *je prechodová funkcia (*δ*-funkcia) a* $h : \mathbb{N} \to \Gamma^*$ *je funkcia, ktorej výstup, t.j.* $h(|w|)$*, kde* w *je vstupný reťazec, je zapísaný na pomocnej páske.*

Rovnosť $\delta(q, a, b) = (q', d_1, d_2)$ znamená, že ak je A v stave q, jeho hlava na vstupnej, resp. pomocnej páske číta symbol a, resp. b, potom v jednom kroku prejde do stavu q' a hlava na vstupnej, resp. pomocnej páske sa posunie o d_1, resp. d_2 pozícií vpravo.

Definícia 4.1.2. ***Konfigurácia*** *DFAT je trojica* $(q, ¢w_1u_{in}w_2\$, ¢v_1u_{adv}v_2\$)$*, pričom* $¢, \$$ *označujú začiatok, resp. koniec pások,* $w_1w_2 \in \Sigma^*$ *je obsah vstupnej pásky,* $v_1v_2 \in \Gamma^*$*, kde* $v_1v_2 = h(|w_1w_2|)$ *je obsah pomocnej pásky,* $q \in K$ *je aktuálny stav automatu a* u_{in}*, resp.* u_{adv} *sú špeciálne znaky pre zaznačenie pozície hlavy na vstupnej, resp. pomocnej páske.*

V dôkazoch sa ale budeme skôr stretávať s tzv. „slabou“ konfiguráciou, kde uvažujeme iba aktuálny stav automatu, pozíciu hláv na vstupnej, resp. pomocnej páske (t.j. indexy políčok) a znaky, ktoré tieto hlavy v danej chvíli vidia.

Definícia 4.1.3. ***Slabá konfigurácia*** *DFAT je trojica* $(q, n_{input}, i_{advice}, x, y)$*, pričom* n_{input}*, resp.* n_{advice} *označujú pozíciu hlavy na vstupnej, resp. pomocnej páske a* x, y *sú znaky, na ktorých sa nachádza hlava na vstupnej, resp. pomocnej páske.*

Krok výpočtu ($\vdash_A$) a tranzitívny uzáver ($\vdash_A^*$) sú pri $DFAT$ definované podobne ako pri deterministických Turingových strojoch s dvomi, resp. viacerými páskami, pričom neuvažujeme blanky a nie je dovolený ani zápis na pásky.

Definícia 4.1.4. ***Jazyk*** *akceptovaný* $DFAT$ $A = (K, \Sigma, \Gamma, \delta, q_0, F, h)$ *je množina* $L(A) = \{w \mid w \in \Sigma^*, \exists q_F \in F : (q_0, ¢u_{in}w\$, ¢u_{adv}h(|w|)\$) \vdash_A^* (q_F, ¢w\$u_{in}, ¢h(|w|)\$u_{adv})\}$.

Ak nebude uvedené inak, predpokladáme, že $DFAT$ bude mať jednu vstupnú pásku a jednu pásku pre pomocnú informáciu a budeme uvažovať binárnu abecedu pre vstup aj pre pomocnú informáciu.

Triedu jazykov akceptovaných automatmi $DFAT$ budeme označovať **DSPACE(1)**$/*$, pričom za lomkou bude obvykle namiesto symbolu $*$ nasledovať obmedzenie pre veľkosť pomocnej informácie, resp. ďalšie upresnenia, ktoré teraz uvedieme.

Pre spôsoby, akými obmedzíme pohyb hláv automatu po páskach, budeme používať nasledujúce označenia:

- **rt-input** - čítanie vstupu v reálnom čase, t.j. hlava na vstupnej páske sa v každom kroku výpočtu musí posunúť vpravo.

- **rt-advice** - čítanie pomocnej informácie v reálnom čase, t.j. hlava na pomocnej páske sa musí posunúť vpravo v každom kroku výpočtu.
- **1w/2w-input** - 1/2-smerný vstup, t.j. či sa hlava môže hýbať po vstupe len jedným alebo obomi smermi.

Ak nebude uvedené inak, uvažujeme $DFAT$ s jednosmernou vstupnou aj pomocnou páskou a v každom kroku výpočtu sa môže pohnúť ľubovoľná z hláv $DFAT$, prípadne obe.

4.1.2 Výsledky

V tejto časti uvedieme aj s dôkazmi niektoré výsledky z článku [KSY13] doplnené o ďalšie naše postrehy.

Veta 4.1.5 ([KSY13])**.** *Ľubovoľný jazyk L sa dá akceptovať pomocou $DFAT$ s jednosmernou vstupnou páskou a dvomi jednosmernými pomocnými páskami.*

Dôkaz. Nech L je ľubovoľný jazyk nad abecedou Σ. Zostrojíme automat A, ktorý rozpoznáva L využijúc jednu jednosmernú vstupnú pásku a dve jednosmerné pomocné pásky.

Nech $\Gamma = \Sigma \cup \{c_a, c_r\}$ je abeceda pomocnej informácie, kde $\Sigma \cap \{c_a, c_r\} = \emptyset$. Myšlienka konštrukcie spočíva v tom, že na prvú pomocnú pásku vypíšeme všetky slová dĺžky n nad abecedou Σ a podľa toho, či dané slovo patrí do L alebo nie, bude za ním nasledovať špeciálny znak c_a, resp. c_r. Formálny zápis vyzerá nasledovne: $w_1 c_1 w_2 c_2 \dots w_{|\Sigma|^n} c_{|\Sigma|^n}$, pričom $w_1, \dots, w_{|\Sigma|^n}$ sú lexikograficky usporiadané všetky slová dĺžky n a $c_i \in \{c_a, c_r\}$ pre $i \in \{1, \dots, |\Sigma|^n\}$ Druhú pomocnú pásku použijeme ako počítadlo, aby sme sa vedeli na zozname slov posunúť o $n+1$ a jej obsah bude vyzerať nasledovne: „$c_a c_r^n \dots c_a c_r^n c_a$“ so $|\Sigma|^n$ opakovaniami podslova $c_a c_r^n$.

Výpočet bude prebiehať tak, že automat bude zo vstupu po jednom čítať znaky. Pri každom prečítanom znaku zo vstupu bude posúvať hlavu na prvej pomocnej páske o $n+1$ znakov doprava (t.j. iterovať po slovách lexikograficky usporiadaných na prvej pomocnej páske), dokým neuvidí zhodný znak s tým na vstupe. Druhú pomocnú pásku automat bude využívať na presun o $n+1$ znakov na prvej pomocnej páske. Toto sa bude opakovať, dokým automat nedočíta celý vstup. Napokon nám stačí zistiť, či na prvej pomocnej páske nasledoval znak c_a alebo c_r a podľa toho automat akceptuje, resp. zamietne vstup. □

4.1.3 Vzťah modelu s pomocnou stopou a DFAT

V nasledujúcej vete porovnáme triedu jazykov akceptovaných deterministickými konečnými automatmi s pomocou informáciou na dodatočnej stope (**REG**/n(*advice track*)) s jazykmi akceptovanými $DFAT$ s jednosmerným vstupom, pričom hlava na páske s pomocnou informáciou sa nemusí pohnúť v každom kroku výpočtu (**DSPACE(1)**/n(*rt-input*)).

Veta 4.1.6 ([KSY13]). ***REG***/n(*advice track*) $\subsetneq$ ***DSPACE(1)***/n(*rt-input*).

Dôkaz. Z definície týchto tried triviálne vyplýva:

$$\begin{aligned}\textbf{REG}/n(\textit{advice track}) &= \textbf{DSPACE(1)}/(\textit{rt-input}, \textit{rt-advice}) \\ &\subseteq \textbf{DSPACE(1)}/n(\textit{rt-input}).\end{aligned}$$

Nech $|w|_\sigma$ označuje počet výskytov znaku σ v slove w. Uvažujme jazyk $EQUAL_{0,1} = \{w \mid w \in \{0,1\}^* \wedge \#_0(w) = \#_1(w)\}$. O tomto jazyku je z vety 3.1.1 známe, že sa nenachádza v triede **REG**/n(*advice track*).

Zostrojíme $DFAT$ A taký, že vstupy nepárnej dĺžky zamietne a pre vstupy párnej dĺžky budeme mať pomocnú informáciu $f(n) = a^{\frac{n}{2}}$, pričom hlava automatu bude postupne prechádzať po vstupnej páske a vždy, keď nájde znak a, posunie sa na pomocnej páske o jeden krok doprava a akceptuje, ak na vstupe naráta presne $\frac{n}{2}$ výskytov znaku a, čo znamená, že na vstupe bol rovnaký počet znakov a a b. □

Veta 4.1.7 ([KSY13]). ***REG***/n(*advice track*) $\subsetneq$ ***DSPACE(1)***/n(1*w-input*, *rt-advice*).

Dôkaz. Z definície týchto tried priamo vyplýva:

$$\begin{aligned}\textbf{REG}/n(\textit{advice track}) &= \textbf{DSPACE(1)}/(\textit{rt-input}, \textit{rt-advice}) \\ &\subseteq \textbf{DSPACE(1)}/n(\textit{rt-input}).\end{aligned}$$

Uvažujme jazyk $EQUAL = \{w \mid w \in \{a,b,c\}^* \wedge \#_a(w) = \#_b(w)\}$. Pre veľmi podobný jazyk vieme z vety 3.1.1, že nepatrí do triedy **REG**/n(*advice track*) a nie je ťažké nahliadnuť, že analogickými úvahami sa dá dokázať, že $EQUAL \notin$ **REG**/n(*advice track*).

Teraz opíšeme konštrukciu $DFAT$ A akceptujúceho jazyk $EQUAL$.

Ako pomocnú informáciu použijeme $f(n) = a^{2n}$. Využijeme skutočnosť, že $2n = |w|_a + 3|w|_b + 2(n - |w|_a - |w|_b)$ práve vtedy, keď $|w|_a = |w|_b$. To znamená, že ak na vstupe hlava prečíta znak a, na pomocnej páske sa hlava posunie o jeden krok doprava. Pokiaľ hlava na vstupe prečíta b, na pomocnej páske sa hlava posunie o tri kroky doprava a v prípade c o dva kroky doprava. Pokiaľ po dočítaní vstupu budú obe hlavy na koncoch pások (znak \$), automat akceptuje. □

Nasledujúca veta hovorí o celej triede jazykov, ktoré sa dajú akceptovať pomocou $DFAT$ s jednosmerným vstupom s pomocnou informáciou viac ako konštantnej dĺžky, ale nie pomocou konečných automatov s pomocnou stopou.

Veta 4.1.8 ([KSY13]). *Pre každú funkciu $f : \mathbb{N} \to \mathbb{N}$ takú, že $f(n) = \omega(1) \cap O(\sqrt{n})$, platí:*

$$\boldsymbol{REG}/n(advice\ track) \subsetneq \boldsymbol{DSPACE(1)}/f^2(n)(1w - input).$$

Dôkaz. Uvažujme jazyk $L_f = \{a^k b^m c^k | k \leq f(n), n = k+m+k\}$, pre ľubovoľnú funkciu f spĺňajúcu predpoklady vety.

O tomto jazyku sa dá pomocou Vety 2 v [Yam10] ukázať, že nepatrí do $\mathbf{REG}/n(advice\ track)$.

Teraz zostrojíme $DFAT$, ktorý bude rozpoznávať L_f. Funkcia pomocnej informácie bude $h(n) = \#a\#aa\#\ldots\#a^{f(n)}\#$ a dĺžka pomocnej informácie bude teda rádovo $O(f^2(n))$. Ak v akomkoľvek momente zistíme, že vstup porušuje tvar $a^*b^*c^*$, tak ho zamietneme.

Pomocnú informáciu využijeme ako počítadlo. Ak hlava prečíta na vstupe znak a, hlava na pomocnej páske sa posunie na nasledujúcu mriežku. Pri prečítaní každého znaku b hlava na pomocnej páske stojí a napokon po každom prečítaní znaku c sa hlava na pomocnej páske pohne o jeden znak doprava.

Na vstupe tvaru $a^*b^*c^*$ teda platí, že v momente, keď dočítame všetky znaky a, nech je ich x, tak hlava na pomocnej páske sa bude nachádzať na úseku tvaru $\ldots\#a^x\#\ldots$, znaky c ignorujeme a potom už len overíme pomocou simultánneho čítania pomocnej informácie a vstupu, či je na vstupe reťazec b^x. □

4.1.4 Vzťah modelu s pomocnou informáciou zreťazenou so vstupom a DFAT

Ďalšia veta hovorí o vzťahu automatov s pomocnou páskou a pomocnou stopou pre konštantnú dĺžku pomocnej informácie. Ukážeme, že konštantná pomocná informácia zreťazená so vstupom je ekvivalentná s pomocnou informáciou na dodatočnej páske, pričom môžeme vstup čítať obomi smermi.

Veta 4.1.9 ([KSY13]).

$$\begin{aligned}\forall k \in \mathbb{N}, \boldsymbol{REG}/k &= \boldsymbol{DSPACE(1)}/k(1w\text{-}input) \\ &= \boldsymbol{DSPACE(1)}/k(2w\text{-}input).\end{aligned}$$

Dôkaz. Inklúzia $\mathbf{REG}/k \subseteq \mathbf{DSPACE(1)}/k(1w\text{-}input) \subseteq \mathbf{DSPACE(1)}/k(2w\text{-}input)$ je zjavná - z automatu používajúceho pomocnú stopu možno priamočiaro vyrobiť automat využívajúci pomocnú pásku bez zmeny nárokov na veľkosť pomocnej informácie.

Bez ujmy na všeobecnosti uvažujme binárnu vstupnú abecedu. Opačná inklúzia vyplýva z toho, že $DFAT$ s obojsmerným vstupom využívajúci k bitov pomocnej informácie zodpovedá pre každú z možných pomocných informácii 2^k dvojsmerným klasickým DKA (značíme $2DKA$), pričom každý z nich sa dá dostať tak, že napevno „zadrôtujeme“ pomocnú informáciu do δ-funkcie. Zároveň je známe, že $2DKA$ sa dajú previesť na ekvivalentné DKA ([RS59]). Ľahko nahliadneme, že takýto DKA nie je problém prispôsobiť, aby využíval k bitov pomocnej informácie zapísanej na dodatočnej stope. □

4.1.5 Jazyk palindrómov

Teraz sa pozrime na vlastnosti jazyka palindrómov párnej dĺžky nad abecedou $\{a, b\}$ (skrátene PAL).

Definícia 4.1.10. $PAL = \{ww^R \mid w \in \{a.b\}^*\}$.

Pri analýze, či sa dá jazyk palindrómov akceptovať na $DFAT$, využijeme vlastnosti relácie ekvivalencie $\equiv_{L,n,k}$ na množine Σ^k, ktorú teraz definujeme a dokážeme o nej lemu 4.1.12.

Definícia 4.1.11 ([KSY13])**.** *Pre ľubovoľnú abecedu Σ, ľubovoľný jazyk $L \subseteq \Sigma^*$ je relácia ekvivalencie $\equiv_{L,n,k}$ na množine Σ^k definovaná nasledovne:*

$$\forall x, y \in \Sigma^k : x \equiv_{L,n,k} y \iff (\forall z \in \Sigma^{n-k}, xz \in L \iff yz \in L)$$

.

Lema 4.1.12 ([KSY13])**.** *Nech f je ľubovoľná funkcia dĺžky pomocnej informácie a nech jazyk $L \in$ **DSPACE(1)**$/f(n)(1w$-input$)$. Potom $\forall n, k \in \mathbb{N}, k \leq n$, platí, že $\equiv_{L,n,k}$ má $O(f(n))$ tried ekvivalencie.*

Dôkaz. Nech A je $DFAT$ akceptujúci jazyk L využívajúci pomocnú informáciu dĺžky $O(f(n))$. Ak zafixujeme pozíciu hlavy na vstupnej páske, existuje len $O(f(n))$ možných kombinácii pozície hlavy na pomocnej páske a stavu. Predpokladajme, že by relácia $\equiv_{L,n,k}$ mala $o(f(n))$ tried ekvivalencie.

Potom by pre dosť veľké n existovali reťazce x a y dĺžky k v dvoch rôznych triedach ekvivalencie také, že automat by sa po ich prečítaní nachádzal v rovnakom stave.

Z toho ale vyplýva, že pre ľubovoľný reťazec $z \in \Sigma^*$, ktorý pripojíme k x, resp. y, čím dostaneme slová xz, resp. yz, bude $DFAT$ A odpovedať rovnako, čo znamená, že by muselo platiť $x \equiv_{L,n,k} y$, čo je spor s tým, že x a y sú v rôznych triedach ekvivalencie. □

Nasledujúca veta hovorí o tom, že na akceptáciu jazyka PAL na $DFAT$ nám nepostačuje pomocná informácia polynomiálnej dĺžky.

Veta 4.1.13 ([KSY13]). $PAL \notin$ ***DSPACE(1)***$/poly(1w\text{-}input)$.

Dôkaz. Uvažujme množinu S všetkých reťazcov dĺžky $k = n/2$ pre párne n nad abecedou $\{a, b\}$. Pre žiadne dva rôzne reťazce x, y neplatí, že $x \equiv_{PAL,n,k} y$, pričom množina S má $2^{\frac{n}{2}}$ prvkov, z čoho na základe lemy 4.1.12 dostávame, že $PAL \notin$ **DSPACE(1)**$/poly$. □

Automaty so vstupom typu real time nemôžu využiť z pomocnej informácie viac ako n znakov, kde n je dĺžka vstupu, z čoho dostávame nasledujúci dôsledok:

Dôsledok 4.1.14. *Pre ľubovoľnú funkciu* $f : \mathbb{N} \to \mathbb{N}$, $PAL \notin$ ***DSPACE(1)***$/f(n)(rt\text{-}input)$

Veta 4.1.15 ([KSY13]). ***DSPACE(1)***$/exp(2w\text{-}input, rt\text{-}advice) = ALL$, *t.j. ak umožníme hlave na vstupnej páske obojsmerný pohyb a využijeme pomocnú informáciu exponenciálnej dĺžky, tak môžeme akceptovať ľubovoľný jazyk.*

Dôkaz. Pomocná informácia pre vstup dĺžky n bude pozostávať zo zreťazenia všetkých slov dĺžky n akceptovaného jazyka, každé z nich oddelené $n + 2$ blankmi. Automat bude čítať vstup a porovnávať ho zaradom so slovami zapísanými na pomocnej páske a vždy, keď zistí, že slovo na vstupe sa nezhoduje s aktuálne čítaným slovom na páske, tak vráti hlavu na vstupnej páske na začiatok a na pomocnej páske prejde za ten čas po blankoch, až na začiatok nasledujúceho slova. Automat akceptuje, keď nájde medzi slovami na pomocnej páske to isté slovo, čo je na vstupe, inak zamietne. □

Veta 4.1.16 ([KSY13]). ***DSPACE(1)***$/poly(1w\text{-}input) \subsetneq$ ***DSPACE(1)***$/poly(2w\text{-}input)$.

Dôkaz. Vo vete 4.1.13 sme ukázali, že na akceptáciu jazyka PAL na $DFAT$ nám nestačí polynomiálna pomocná informácia v prípade, že máme jednosmernú hlavu na vstupnej páske. V prípade, ak je hlava obojsmerná, tak jazyk PAL je možné akceptovať využijúc kvadratickú pomocnú informáciu nasledujúcim spôsobom:

Ak je n nepárne, tak vstup zamietneme. Ak je n párne, tak ako pomocnú informáciu použijeme zoznam bitových masiek všetkých dvojíc pozícií vo vstupnom slove, ktoré treba porovnať, teda $\forall i \in \{1..n\} : (i, n - i + 1)$. Jednotlivé masky budú oddelené znakom mriežky.

Napríklad pre slová dĺžky 6 bude vyzerať zoznam masiek kódujúcich porovnávané dvojice znakov takto:

$$h(6) = \#100001\#010010\#001100\#.$$

Hlava sa bude naraz pohybovať po vstupnej aj po pomocnej páske a keď natrafí na pomocnej páske na znak 1, tak si do stavu uloží aktuálne porovnávaný znak a porovná ho, keď na pomocnej páske uvidí ďalšiu jednotku. Keď príde na pomocnej páske na mriežku, tak hlava na vstupnej páske prejde späť na začiatok a tento proces sa postupne opakuje pre všetkých $n/2$ porovnávaných dvojíc. Ak nikde porovnávanie nezlyhá, automat akceptuje. Využili sme obojsmernosť hlavy na vstupe a dĺžka využitej pomocnej informácie je $\frac{n^2}{2} + O(n)$, čiže je kvadratická. □

V článku [KSY13] ostalo otvoreným problémom, či sa dá na $DFAT$ akceptovať jazyk PAL s exponenciálnou pomocnou informáciou ([KSY13], strana 7), resp. či **DSPACE(1)**/(1*w-input*) obsahuje v sebe všetky jazyky. My sme sa pozreli na v istom zmysle súvisiaci problém a to, akú najdlhšiu pomocnú informáciu má zmysel uvažovať pri $DFAT$.

Veta 4.1.17. *Nech L je ľubovoľný jazyk z **DSPACE(1)**/$*$ nad binárnou abecedou a s je počet stavov $DFAT$ A, ktorý ho akceptuje. Potom platí, že $L \in$ **DSPACE(1)**/$l \cdot (n \cdot 2^n + 1)$, kde $l = s^{2s} + 1$.*

Dôkaz. Ak A využíva pomocnú informáciu dĺžky najviac $l \cdot (n \cdot 2^n + 1)$ pre vstupy dĺžky n, potom veta platí. Predpokladajme teda, že pre nejaké n, A využíva pomocnú informáciu dĺžky viac ako $l \cdot (n \cdot 2^n + 1)$. Ukážeme, že túto pomocnú informáciu je možné skrátiť na dĺžku najviac $l \cdot (n \cdot 2^n + 1)$.

Pre vstup dĺžky n máme 2^n možností ako môže vstupné slovo vyzerať. Rozdeľme obsah pomocnej pásky na bloky dĺžky l.

Pozrime sa teraz na každé z 2^n možných vstupných slov, konkrétne sledujme, ako sa hýbu hlavy na vstupe, resp. pomocnej páske.

Intuitívne - bloky, kde sa pohne hlava na vstupnej aj pomocnej páske, sú pre výpočet relevantnejšie ako bloky, kde sa hlava bude hýbať iba na pomocnej páske, a na vstupnej sa nepohne.

Označme si každý blok, pre ktorý existuje nejaké vstupné slovo také, že keď cez tento blok prechádzala hlava na pomocnej páske, tak aj hlava na tomto vstupnom slove sa aspoň raz pohla. Označené bloky zodpovedajú tým intuitívne „relevantným". Zrejme takýchto blokov môže byť maximálne $n \cdot 2^n$, t.j. na každý pohyb hlavy na danom vstupnom slove môže pripadať iba jeden označený blok.

Náš predpoklad je, že na páske je viac ako $n \cdot 2^n$ blokov, máme teda istotu, že existuje aspoň jeden neoznačený blok.

Vyberme teda ľubovoľný z neoznačených blokov, označme ho B a zamerajme sa teraz na to, ako tento blok môžeme skrátiť. V tomto bode zohrá dôležitú úlohu, že dĺžka bloku je práve $l = s^{2s} + 1$.

Vizualizujme si postupnosti stavov automatu na bloku B, pričom v prvom stĺpci máme zapísané všetky možné dvojice $(znak, stav)$ tvorené znakom, ktorý vidí hlava na nejakom vstupe dĺžky n v momente, keď hlava na pomocnej páske vstúpi do bloku B a stavom, v ktorom sa nachádza automat v tomto momente. Následne v $(i+1)$-vom stĺpci je zaznačený stav automatu pri vstupe na i-te políčko bloku B, pričom postupnosť takýchto stavov v danom riadku zodpovedá dvojici $(znak, stav)$, ktorá je na začiatku tohto riadku.

	↓		↓	
(a, q_0)	**q_3**	**q_1**	**q_1**	**....**
(a, q_1)	**q_9**	**q_2**	**q_2**	**....**
.	.	.	.	
.	.	.	.	
.	.	....	.	....
(b, q_0)	**q_1**	**q_0**	**q_0**	**....**
(b, q_1)	**q_7**	**q_5**	**q_5**	**....**
.	.	.	.	
.	.	.	.	
.	.	....	.	....

Obr. 4.1: Príklad - vypísané všetky možné prechodové postupnosti na danom bloku pomocnej informácie podľa toho, aký znak číta hlava na vstupnej páske a v akom stave je automat v momente, keď hlava na pomocnej páske vstúpi do bloku B (prvý stĺpec). Ďalšie stĺpce zodpovedajú postupnosti stavov dosahovaných na jednotlivých políčkach bloku B. Šípkou je označená dvojica rovnakých stĺpcov.

Táto tabuľka (obrázok 4.1) má $2s$ riadkov, kde 2 je veľkosť abecedy a s je počet stavov automatu. Naším cieľom je skrátiť blok B pomocou tzv. „cut and paste“ metódy, ktorú si teraz podrobnejšie opíšeme.

Keďže dĺžka bloku je $l = s^{2s} + 1$, čo je zároveň počet stĺpcov (okrem prvého) našej tabuľky a na druhe strane, počet všetkých možností, ako stĺpec v tabuľke (okrem prvého) môže vyzerať, je s^{2s}, z Dirichletovho princípu dostávame, že v tabuľke z obrázku 4.1 musí existovať dvojica rovnakých stĺpcov zodpovedajúca dvom rôznym pozíciam hlavy vrámci bloku B. Nech sú to stĺpce s indexmi i a j.

Zoberme ľubovoľné vstupné slovo w dĺžky n. Bez ujmy na všeobecnosti predpokladajme, že pri každom výpočte, hlava na pomocnej páske $DFAT$ A prejde celú pomocnú informáciu, a teda niekedy vstúpi na blok B. Inak by sme mohli upraviť automat A tak, aby bezprostredne pred akceptovaním/zamietnutím vstupu presunul hlavu na pomocnej páske celkom vpravo, ak tam už nebola.

Nech teda $DFAT$ A pri výpočte na slove w vstúpi v niektorom okamihu na blok

B, pričom A je vtedy v stave q a hlava na vstupe vtedy číta symbol d. Na základe spôsobu, akým sme blok B zvolili vieme, že hlava na vstupe sa nepohne, kým hlava na pomocnej páske sa bude nachádzať v bloku B. Nech pri čítaní bloku B prejde $DFAT$ A postupnosťou stavov $q_1, \ldots, q_i, \ldots, q_j, \ldots q_{s^{2s}+1}$. Táto postupnosť je zrejme v tom riadku tabuľky pre blok B, ktorý sa začína dvojicou (d, q), pričom vieme, že i-ty a j-ty stav sú rovnaké, nakoľko zodpovedajú dvojici rovnakých stĺpcov v tabuľke pre blok B.

Podobne, i-ty a j-ty stav bude rovnaký v postupnosti stavov pre ľubovoľné slovo dĺžky n, a teda sme pre všetky slová dĺžky n našli rovnaký úsek výpočtu, ktorý tvorí slučku. O túto slučku môžeme každý z výpočtov skrátiť, bez toho, aby sme zmenili jeho výsledok, a teda môžeme blok B skrátiť o $j-i > 0$ políčok, čím sme ukázali, že pomocnú informáciu dĺžky prevyšujúcej $l \cdot (n \cdot 2^n + 1)$ je možné skrátiť. Tento proces skracovania možno opakovať, až kým dĺžka pomocnej informácie je najviac $l \cdot (n \cdot 2^n + 1)$. □

Podobným spôsobom môžeme dokázať zovšeobecnenie vety 4.1.17 pre k-árne abecedy.

Veta 4.1.18. *Nech L je ľubovoľný jazyk z **DSPACE(1)** nad k-árnou abecedou a s je počet stavov $DFAT$ A, ktorý ho akceptuje. Potom platí, že $L \in$ **DSPACE(1)**$/l \cdot (n \cdot k^n + 1)$, kde $l = s^{ks} + 1$.*

4.2 Nedeterministické konečné automaty s pomocnou páskou

V článku [KSY13] sa autori venovali predovšetkým deterministickým a pravdepodobnostným konečným automatom s pomocnou páskou. V tejto časti skonfrontujeme známe výsledky o deterministikých automatoch s našimi výsledkami o nedeterministických automatoch.

4.2.1 Definície

Najprv definujme, čo znamená nedeterminizmus pri konečných automatoch s pomocnou páskou.

Definícia 4.2.1. ***Nedeterministický konečný automat s pomocnou páskou (skrátene NFAT)*** *A je sedmica $(K, \Sigma, \Gamma, \delta, q_0, F, h)$, kde K je konečná množina stavov, Σ je konečná vstupná abeceda, Γ je konečná abeceda pomocnej informácie, $q_0 \in K$ je počiatočný stav, $F \subseteq K$ je množina (konečných) akceptačných stavov, $\delta : K \times (\Sigma \cup \{\epsilon, ¢, \$\}) \times (\Gamma \cup \{\epsilon, ¢, \$\}) \to 2^{K \times \{0,1\} \times \{0,1\}}$ je prechodová funkcia (δ-funkcia) a $h : \mathbb{N} \to \Sigma^*$*

je funkcia, ktorej výstup, t.j. $h(|w|)$, kde w je vstupný reťazec, je zapísaný na pomocnej páske.

Ak nebude uvedené inak, predpokladáme, že $NFAT$ bude mať jednu vstupnú pásku a jednu pásku pre pomocnú informáciu a budeme uvažovať binárnu abecedu pre vstup aj pre pomocnú informáciu.

Krok výpočtu funguje pri $NFAT$ podobne ako pri nedeterministických Turingových strojoch s dvomi resp. viacerými páskami, pričom neuvažujeme blanky a nie je dovolený ani zápis na pásky.

Konfigurácia, resp. „slabá" konfigurácia a jazyk akceptovaný $NFAT$ sú definované podobne ako pre $DFAT$.

Triedu jazykov, pre ktoré existuje $NFAT$, ktorý ich akceptuje, budeme označovať **NSPACE(1)**/$*$, pričom za lomkou bude obvykle namiesto symbolu $*$ nasledovať obmedzenie pre veľkosť pomocnej informácie, resp. ďalšie obmedzenia, podobne ako v deterministickom prípade.

4.2.2 Výsledky

V tejto časti uvedieme naše výsledky ohľadom nedeterministických konečných automatov s pomocnou páskou ($NFAT$). Definícia $NFAT$ je podobná definícií $DFAT$, jediný rozdiel je v prechodovej funkcii, ktorá umožňuje automatu vybrať si z viacerých možností a v definícii akceptačného výpočtu, podobne ako je tomu pri klasických DKA a NKA. Modely $NFAT$ a $DFAT$ budeme vzájomne porovnávať vzhľadom na obmedzenú dĺžku pomocnej informácie, obvykle polynomiálnu a väčšinou budeme uvažovať v oboch prípadoch model hlavy, ktorá sa môže hýbať síce len jedným smerom, ale rôznymi rýchlosťami po vstupnej a pomocnej páske. Začneme niekoľkými, zväčša triviálnymi, postrehmi.

Veta 4.2.2. *Na akceptovanie ľubovoľného jazyka L nad abecedou Σ na $NFAT$ stačí pomocná informácia veľkosti najviac $(n+1) \cdot |\Sigma|^n$, kde n je dĺžka vstupu.*

Dôkaz. Majme ľubovoľný jazyk L. Zostrojíme k nemu $NFAT$ A taký, že ho akceptuje. Za pomocnú informáciu zvolíme zoznam všetkých slov dĺžky n, ktorých je najviac $|\Sigma|^n$. Zoznam slov bude zapísaný v tvare $\#w_1\#w_2\#\ldots\#w_k$, pričom $w_1, \ldots, w_k \in L$, všetky sú dĺžky n a navzájom rôzne a k je počet slov dĺžky n v L. Hlava na pomocnej páske bude najprv robiť kroky doprava, až zastane na niektorej z mriežok - na ktorej, to si zvolí nedeterministicky, a potom A overí rovnosť reťazca na pomocnej páske medzi dvomi mriežkami a reťazca na vstupe.

Ľahko nahliadneme, že dĺžka pomocnej informácie je najviac $(n+1) \cdot |\Sigma|^n$. □

4.2.3 NFAT a konštantná pomocná informácia

Veta 4.2.3. $\forall k \in \mathbb{N} : \boldsymbol{NSPACE(1)}/k = \boldsymbol{REG}/k = \boldsymbol{DSPACE(1)}/k$

Dôkaz. Rovnosť $\mathbf{REG}/k = \mathbf{DSPACE(1)}/k$ bola dokázaná vo vete 4.1.9.
Inklúzia $\mathbf{NSPACE(1)}/k \supseteq \mathbf{REG}/k$ je zjavná, ostáva teda ukázať opačnú inklúziu, t.j. $\mathbf{NSPACE(1)}/k \subseteq \mathbf{REG}/k$. Nech A je ľubovoľný $NFAT$ pracujúci s konštantnou pomocnou informáciou. Všimnime si, že A môžeme previesť na ekvivalentný $NFAT$ A', taký, že najprv si prečíta pomocnú informáciu, uloží si ju do pamäte, resp. do stavu a od tohto bodu ďalej simuluje A. Takýto $NFAT$ môžeme potom priamočiaro previesť na klasický NKA, ktorý dostane pomocnú informáciu zreťazenú so vstupom. Tento NKA môžeme potom previesť štandardnou konštrukciou (uvedená napr. v [HMU01], veta 2.1) na ekvivalentný DKA, čím sa nám podarilo nájsť spôsob, ako pomocou deterministického automatu s pomocnou informáciou zreťazenou so vstupom odsimulovať $NFAT$.

□

4.2.4 Jazyk palindrómov a uzavretosť vzhľadom na komplement

Veta 4.2.4. *Neexistuje $NFAT$ s jednosmernou hlavou na vstupnej a pomocnej páske s polynomiálnou pomocnou informáciou akceptujúci jazyk palindrómov PAL.*

Dôkaz. Predpokladajme, že by existoval taký $NFAT$ A pre jazyk PAL.

Palindrómov dĺžky n nad binárnou abecedou je $2^{\frac{n}{2}}$, čiže exponenciálne veľa. Po prečítaní prvej polovice vstupu dĺžky n môže byť automat len v polynomiálne veľa rôznych konfiguráciách, nakoľko dĺžka pomocnej informácie je polynomiálna, pričom za konfiguráciu pre účely tohto dôkazu považujeme slabú konfiguráciu.

Pre každý reťazec ww^R dĺžky n z jazyka PAL vyberme ľubovoľný akceptačný výpočet a nech c_w je konfigurácia, v ktorej sa nachádza A po prečítaní prvej polovice vstupu ww^R vo vybranom akceptačnom výpočte.

Z Dirichletovho princípu teda vyplýva, že pre dostatočne veľké n existujú dva rôzne reťazce dĺžky n, nech sú to $w_1w_1^R$ a $w_2w_2^R$, pričom $c_{w_1} = c_{w_2}$. Ak vezmeme reťazec $w_1w_2^R$, tento zjavne nebude palindróm, pričom náš $NFAT$ A pre PAL by toto slovo akceptoval, nakoľko $c_{w_1} = c_{w_2}$, čo je spor. □

Veta 4.2.5. *Jazyk $coPAL = \{a, b\}^* - PAL$ sa dá akceptovať pomocou $NFAT$ s jednosmernou hlavou na vstupnej a pomocnej páske využívajúceho pomocnú informáciu dĺžky $\frac{n(n+1)}{2}$.*

Dôkaz. Náš $NFAT$ A akceptuje $coPAL$ nasledovne.

Ak je vstup nepárnej dĺžky, tak môžeme predpokladať, že $NFAT$ A si túto skutočnosť nedeterministicky tipne, overí, že vstup je nepárnej dĺžky a akceptuje.

V prípade párnej dĺžky vstupu pomocná informácia pre vstupy dĺžky n bude vyzerať rovnako ako v dôkaze vety 4.1.16, čiže budú tam zreťazené pomocou mriežok všetky palindrómy dĺžky n nad abecedou $\{0,1\}$ obsahujúce práve dve jednotky - tých je $\frac{n}{2}$ a dĺžka pomocnej informácie bude preto dokopy $\frac{n(n+1)}{2}$. Tieto jednotky budú označovať pozície, ktoré budeme porovnávať.

Hlava automatu na pomocnej páske si nedeterministicky vyberie, podľa ktorého z týchto slov zreťazených mriežkami bude hlava na vstupe porovnávať a ak zistí, že dvojica porovnávaných znakov na vstupe sa nezhoduje, tak akceptuje. □

Z vety 4.2.4 a 4.2.5 vyplýva tento dôsledok:

Dôsledok 4.2.6. *Trieda jazykov akceptovaných $NFAT$ s pomocnou informáciou polynomiálnej dĺžky a jednosmernou hlavou nie je uzavretá vzhľadom na komplement.*

4.3 Hierarchia jazykov akceptovaných na $DFAT$ a $NFAT$

V tejto časti sa pozrieme na rôzne hierarchie nad jazykmi akceptovanými $NFAT$, resp. $DFAT$ vzhľadom na dĺžku pomocnej informácie.

4.3.1 Polynomiálna hierarchia pre jazyky akceptované na $DFAT$

V článku [KSY13] autori načrtli polynomiálnu hierarchiu pre jednosmerné $DFAT$. Našli triedu jazykov, o ktorej dokázali nasledujúce tvrdenie:

Veta 4.3.1 ([KSY13])**.** $\forall k \in \mathbb{Z}^+$, ***DSPACE(1)***$/n^k(1w\text{-}input) \subsetneq$ ***DSPACE(1)***$/n^{k+1}(1w\text{-}input)$.

Aby sme dokázali túto vetu, najprv definujeme triedu jazykov L_k pre $k \in \mathbb{Z}^+$ a následne ukážeme, že na ich akceptáciu je nutná a zároveň postačujúca pomocná informácia dĺžky $O(n^i)$ pre ľubovoľné L_i.

Definícia 4.3.2. $\forall k \in \mathbb{Z}^+, L_k = \{c_k^{n_k} c_{k-1}^{n_{k-1}} \dots c_1^{n_1} c_0^{n_0} c_1^{n_1} \dots c_{k-1}^{n_{k-1}} c_k^{n_k} \mid n_0 > 0 \wedge n_j \geq 0$ *pre* $j \in \{1, \dots, k\}\}$ *na* $k+1$ *znakovej abecede* $\{c_0, c_1, \dots, c_k\}$.

Lema 4.3.3 ([KSY13])**.** $\forall i \in \mathbb{Z}^+, L_i \notin DSPACE(1)/n^{i-1}(1w\text{-}input)$.

Dôkaz. Nech n je uvažovaná dĺžka slov z L_i. Bez ujmy na všeobecnosti, nech n je párne.

Uvažujme množinu $S_{i,\frac{n}{2}}$ reťazcov tvaru $c_i^* c_{i-1}^* \dots c_1^* c_0^+$ a dĺžky $n/2$. Každý z týchto reťazcov predstavuje možnú prvú polovicu slova z jazyka L_i. Zároveň pre žiadne dva

rôzne reťazce v S neplatí $x \equiv_{L_i,n,\frac{n}{2}} y$, lebo platí $xx^R \in L_i$, resp. $yy^R \in L_i$, a zároveň $xy^R \notin L_i$, $yx^R \notin L_i$.

Počet reťazcov v množine $S_{i,n/2}$ sa dá zdola ohraničiť ako $(\frac{n}{2i})^i = \Theta(n^i)$.

Myšlienka tohto odhadu je, že každému zo znakov $c_0, \ldots, c_i$ v slove tvaru $c_i^* c_{i-1}^* \ldots c_1^*$ umožníme, aby sa zopakoval maximálne $\frac{n}{2(i+1)}$-krát, čím dostaneme slovo tvaru $c_i^* c_{i-1}^* \ldots c_1^*$, ktoré nebude dlhšie ako $\frac{n}{2}$. Zjavne slovo, ktoré takto dostaneme, môžeme potom doplniť na dĺžku $n/2$ tak, že pripojíme k slovu potrebný počet znakov c_0. Týmto sme opísali postup, ako vyrobíme $(\frac{n}{2i})^i$ rôznych slov z množiny $S_{i,n/2}$, čo je hľadaný dolný odhad.

Dostávame teda, že $\equiv_{L_i,n,\frac{n}{2}}$ má $\Theta(n^i)$ tried ekvivalencie, a teda z lemy 4.1.12 dostávame, že $L_i \notin \mathbf{DSPACE(1)}/n^{i-1}(1w\text{-}input)$. □

V nasledujúcej leme ukážeme, že pomocná informácia veľkosti $O(n^k)$ je zároveň postačujúca pre jazyk L_k, aby ho bolo možné akceptovať na $DFAT$ s jednosmernou vstupnou aj pomocnou páskou.

Lema 4.3.4 ([KSY13]). *Pre všetky $i \in \mathbb{Z}^+, L_i \in$ **DSPACE(1)**$/n^i(1w\text{-}input)$.*

Dôkaz. Pomocnú informáciu budeme konštruovať induktívne. Pozrime sa najprv na prípad pre L_1.

Dokážeme, že $L_1 \in \mathbf{DSPACE(1)}/n(1w\text{-}input)$ nájdením vhodnej funkcie pre pomocnú informáciu a zostrojením $DFAT$ A_1, ktorý túto pomocnú informáciu využije na akceptovanie L_1.

Indukčný predpoklad ($i = 1$):

Uvažujme funkciu pre pomocnú informáciu $h_1(n) = 1^n$. Automat overí, že vstup je tvaru $c_1^i c_0^j c_1^k, i, k \geq 0, j > 0$, ak nie, tak zamietne.

Zároveň bude automat pohybovať hlavou po pomocnej páske nasledovným spôsobom:

pre všetky c_1 pred c_0 na vstupe hlava na pomocnej páske stojí, za každé c_0 urobí jeden krok doprava a za každé c_1 po c_0 zasa dva kroky doprava. Automat akceptuje, ak po dočítaní vstupu hlava na oboch páskach bude na konci. Na vstupnej páske urobí hlava $n = i + j + k$ krokov, na pomocnej páske urobí zasa $j + 2k = n$ krokov. Z toho vyplýva, že ak automat akceptoval, muselo platiť $i = k$. Z toho máme, že $DFAT$ A_1 akceptuje jazyk L_1.

Teraz dokážeme, že táto konštrukcia sa dá zovšeobecniť pre ľubovoľné L_i, a to tak, že dokážeme implikáciu

$L_i \in \mathbf{DSPACE(1)}/n^i(1w\text{-}input) \implies L_{i+1} \in SPACE(1)/n^{i+1}(1w\text{-}input)$.

Indukčný krok:

Predpokladajme, že $L_i \in \mathbf{DSPACE(1)}/n^i(1w\text{-}input)$, z čoho vyplýva, že existuje $DFAT$ A_i akceptujúci L_i využívajúci pomocnú informáciu $h_i(n)$ dĺžky $O(n^i)$. Túto

skutočnosť využijeme, aby sme zostrojili $DFAT$ A_{i+1} akceptujúci L_{i+1} s pomocnou funkciou $h_{i+1}(n)$, vyzerajúcou nasledovne:

$h_{i+1}(n) = h_i(n)\#_{i+1}h_i(n-2)c_{i+1}\#_{i+1}\ldots\#_{i+1}h_i(n-2\lfloor n/2\rfloor)c_{i+1}^{\lfloor n/2\rfloor}\#_{i+1}$

pričom $\#_{i+1}$ je nový symbol dodaný do abecedy pomocnej informácie. Z konštrukcie vidno, že ak dĺžka $h_i(n)$ je $O(n^i)$, tak $h_{i+1}(n)$ bude mať dĺžku $O(n^{i+1})$.

$DFAT$ A_{i+1} bude fungovať tak, že paralelne bude overovať, či je vstup tvaru $c_{i+1}^*c_i^*\ldots c_1^*c_0^+c_1^*\ldots c_i^*c_{i+1}^*$ a zároveň pre každé c_{i+1} na vstupe nachádzajúce sa pred c_i sa pohne hlava na pomocnej páske na nasledujúcu $\#_{i+1}$. Keď hlava na vstupnej páske uvidí prvý symbol rôzny od c_{i+1}, tak začne simulovať A_i pre L_i s príslušnou pomocnou informáciou, ktorý bude bežať buď dokým hlava na vstupnej páske neprečíta opäť c_{i+1} alebo zistí, že je na konci vstupu. Následne v prípade, že simulovaný A_i akceptoval, tak A_{i+1} porovná počet zvyšných c_{i+1} symbolov na vstupnej a na pomocnej páske a ak sa rovnajú, tak A_{i+1} akceptuje, inak zamietne. □

4.3.2 Hustejšie hierarchie pre jazyky akceptované na $DFAT$ a $NFAT$

Podobný druh hierarchie, dokonca hustejšej ako vo vete 4.3.4, sa nám podarilo nájsť aj pre jazyky rozpoznávané jednosmernými $NFAT$. Základom budú úvahy podobné tým, ktoré boli použité v dôkaze vety 4.1.13.

Veta 4.3.5. *Pre funkcie $f, g : \mathbb{N} \to \mathbb{N}$ také, že $f(n)\cdot log(f(n)) = o(g(n))$ a $g(n) \leq n2^{\frac{n}{2}}$, platí: **NSPACE(1)**$/f(n) \subsetneq$ **NSPACE(1)**$/g(n)$.*

Dôkaz. Uvažujme jazyk $PAL_{g(n)} = \{ww^Rv \mid w \in \{a,b\}^m$, m je maximálne nezáporné celé číslo také, že: $2^m \cdot (2m+1) \leq g(n), v = b^{n-2m}\}$. Keďže platí $(2m+1)2^m \leq g(n)$ a podľa predpokladu tiež $g(n) \leq n2^{\frac{n}{2}}$, musí tiež platiť $2m \leq n$. To znamená, že podslová $v = b^{n-2m}$ slov z jazyka $PAL_{g(n)}$ sú korektne definované.

Dokážeme najprv, že $PAL_{g(n)} \notin$ **NSPACE(1)**$/f(n)(1w\text{-}input)$.

Predpokladajme, že by existoval $NFAT$ A s pomocnou informáciou veľkosti najviac $cf(n)$ (pre nejaké $c \geq 1$) akceptujúci jazyk $PAL_{g(n)}$. Nech q je počet stavov automatu A. Nech n je ľubovoľné prirodzené číslo a nech m_n je maximálne nezáporné celé číslo také, že

$$2^{m_n}(2m_n+1) \leq g(n), \tag{4.1}$$

t.j. pre číslo m_n+1 platí

$$g(n) < 2^{m_n+1}(2(m_n+1)+1). \tag{4.2}$$

Nech

$$d = max\left\{\frac{3}{2}, 8log(qc) - \frac{1}{2}\right\}. \tag{4.3}$$

Pre skoro všetky (t.j. pre všetky, okrem konečného počtu) prirodzené čísla n platí

$$m_n \geq d, \tag{4.4}$$

lebo podľa 4.2 pre všetky n platí

$$2^{m_n+1}(2(2m_n+1)+1) > g(n)$$

a pre skoro všetky n platí

$$g(n) \geq 2^{d+1}(2(d+1)+1),$$

lebo $g(n)$ je zhora neohraničená a neklesajúca.

Dokážeme teraz sporom, že pre skoro všetky prirodzené čísla n platí $qcf(n) < 2^{m_n}$. Nech by teda pre nekonečne veľa n platilo

$$qcf(n) \geq 2^{m_n} \tag{4.5}$$

Preto $log(f(n)) \geq m_n - log(qc)$, a teda

$$\begin{aligned}
f(n)log(f(n)) &\geq f(n)(m_n - log(qc)) \\
&\geq \frac{2^{m_n}m_n}{qc} - \frac{2^{m_n}log(qc)}{qc} && \text{(podľa (4.5))} \\
&= \frac{2^{m_n+1}\cdot 4m_n}{8qc} - \frac{2^{m_n}log(qc)}{qc} \\
&\geq \frac{2^{m_n+1}(2(m_n+1)+1)}{8qc} - \frac{2^{m_n}log(qc)}{qc} && \text{(podľa (4.4) pre } \frac{3}{2}) \\
&> \frac{g(n)}{8qc} - \frac{2^{m_n}log(qc)}{qc} && \text{(podľa (4.2))} \\
&= \frac{g(n)}{16qc} + \left(\frac{g(n)}{16qc} - \frac{2^{m_n}log(qc)}{qc}\right) \\
&\geq \frac{g(n)}{16qc}. && \text{(podľa (4.1) a (4.4) pre } 8log(qc) - \frac{1}{2})
\end{aligned}$$

To by ale znamenalo, že $f(n)log(f(n)) \geq \frac{g(n)}{16qc}$ pre nekonečne veľa n, čo je spor s predpokladom, že $f(n)log(f(n)) = o(g(n))$. Preto pre skoro všetky n platí $qcf(n) < 2^{m_n}$.

Nech n je ľubovoľné prirodzené číslo, pre ktoré $qcf(n) < 2^{m_n}$. Počet slov dĺžky n v jazyku $PAL_{g(n)}$ je 2^{m_n}, lebo každé z nich je jednoznačne určené jeho prefixom dĺžky m_n a počet takýchto rôznych prefixov je 2^{m_n}. Na druhej strane, počet slabých konfigurácií, do ktorých sa A môže dostať na takýchto slovách dĺžky n, keď hlava na vstupe prečíta prefix dĺžky m_n, je $qcf(n) < 2^{m_n}$.

Preto možno dokázať (podobne ako v dôkaze vety 4.2.4), že v $PAL_{g(n)}$ existujú dve rôzne slová $w_1w_1^Rb^{n-m_n}$ a $w_2w_2^Rb^{n-m_n}$ dĺžky n, kde $w_1, w_2 \in \{a,b\}^{m_n}$, a k nim príslušné dva akceptačné výpočty také, že A je v oboch týchto výpočtoch v rovnakej konfigurácií, keď hlava na vstupe prečíta prefix dĺžky m_n. Preto bude A akceptovať aj slovo $w_1w_2^Rb^{n-m_n}$, ktoré ale nepatrí do $PAL_{g(n)}$ a teda $PAL_{g(n)} \notin NSPACE(1)/f(n)$.

Zároveň $PAL_{g(n)} \in \mathbf{NSPACE(1)}/g(n)$, keďže nám stačí na pomocnej páske vypísať si všetky palindrómy nad abecedou $\{a,b\}$ dĺžky $2m_n$ oddelené napríklad mriežkou, čo je reťazec dĺžky $2^{m_n}(2m_n+1) \leq g(n)$. Našli sme teda hľadanú hierarchiu. □

4.3.3 Hierarchia pre jazyky akceptované NFAT a DFAT so sublineárnou pomocnou informáciou

Pre jazyky, na ktoré stačí na $NFAT$, resp. $DFAT$ pomocná informácia dĺžky najviac n, sme našli ešte hustejšiu hierarchiu, o ktorej hovorí nasledujúca veta.

Veta 4.3.6. *Nech $f, g : \mathbb{N} \to \mathbb{N}$ je dvojica funkcií, pre ktoré platí $\forall n \in \mathbb{N} : f(n) \leq n, g(n) = o(f(n))$. Potom existuje jazyk $L_{f,g} \in \{0,1\}^*$ taký, že $L_{f,g} \in DSPACE(1)/f(n)$ a zároveň $L \notin NSPACE(1)/g(n)$.*

Dôkaz. Definujme postupnosť $S = A_1, A_2, A_3, \ldots$ automatov $NFAT$ takú, že sú v nej zoradené všetky možné $NFAT$, ale bez toho, aby sme brali do úvahy funkciu pomocnej informácie. Takáto postupnosť má spočítateľne veľa členov, keďže kódov $NFAT$ bez uvažovania pomocnej informácie je spočítateľne veľa, lebo prechodová funkcia, počiatočný stav a množina konečných stavov sa dajú jednoznačne zakódovať ako binárny reťazec a tomu môžeme priamočiaro priradiť prirodzené číslo.

Zostrojme jazyk $L_{f,g}$ tak, aby pre každé n obsahoval práve jedno slovo dĺžky n tvaru $x0^{n-f(n)}$, kde $x \in \{0,1\}^{f(n)}$. Teda $L_{f,g} \in DSPACE(1)/f(n)$, keďže automat dokáže s pomocnou informáciou x dĺžky $f(n)$ rozpoznať vstup $x0^{n-f(n)}$ dĺžky n. Nech n je ľubovoľné prirodzené číslo. Uvažujme dva prípady.

V prvom prípade pre n neexistuje prirodzené číslo d také, že $(d+d^2)g(n) < f(n)$. V tomto prípade dáme do $L_{f,g}$ slovo $1^{f(n)}0^{n-f(n)}$.

V druhom prípade, nech d je maximálne prirodzené číslo také, že pre n platí $(d + d^2)g(n) < f(n)$. V tomto prípade dáme do $L_{f,g}$ slovo w_n dĺžky n, ktoré nájdeme nasledovne.

Nech A je ľubovoľný $NFAT$, ktorý sa nachádza medzi prvými $2^{dg(n)}$ automatmi v postupnosti S, nech A má k-árnu abecedu pre pomocnú pásku, kde $log_2(k+1) \leq d$ a nech v je ľubovoľné slovo nad touto k-árnou abecedou dĺžky najviac $dg(n)$, ktorú môže A použiť ako pomocnú informáciu.

Budeme hovoriť, že dvojica (A, v) je jednoslovná pre vstupy dĺžky n, ak A akceptuje s pomocnou informáciou v práve jedno slovo dĺžky n a toto je tvaru $x0^{n-f(n)}$, kde $x \in \{0,1\}^{f(n)}$. Pre každý automat A, ktorý sa nachádza medzi prvými $2^{dg(n)}$ automatmi postupnosti S, existuje najviac $2^{d^2g(n)}$ potenciálnych slov v takých, že dvojica (A, v) je jednoslovná pre vstupy dĺžky n, lebo počet slov dĺžky najviac $dg(n)$ nad k-árnou abecedou, kde $log_2(k+1) \leq d$, je najviac $(k+1)^{dg(n)} = (2^{log_2(k+1)})^{dg(n)} \leq 2^{d^2g(n)}$. Keďže takýchto automatov A je najviac $2^{dg(n)}$, celkový počet jednoslovných dvojíc pre vstupy dĺžky n je najviac $2^{dg(n)} \cdot 2^{d^2g(n)} = 2^{(d+d^2)g(n)} < 2^{f(n)}$, lebo $(d + d^2)g(n) < f(n)$.

Ale počet slov dĺžky n tvaru $x0^{n-f(n)}$, kde $x \in \{0,1\}^{f(n)}$, je $2^{f(n)}$, z čoho dostaneme, že medzi slovami dĺžky n uvedeného tvaru musí byť slovo (označme ho w_n), pre ktoré neexistuje žiadna jednoslovná dvojica (A', v') pre vstupy dĺžky n taká, že A' akceptuje w_n s pomocou slova v'. Slovo w_n dáme do jazyka $L_{f,g}$.

Dokážeme teraz sporom, že $L_{f,g} \notin NSPACE(1)/g(n)$. Predpokladajme, že by existoval $NFAT$ $\bar{A}$, ktorý by akceptoval jazyk $L_{f,g}$ s pomocnou informáciou dĺžky najviac $cg(n)$, pre nejaké $c \geq 1$. Nech sa $\bar{A}$ vyskytuje na j-tom mieste v postupnosti S a nech $\bar{A}$ má $\bar{k}$-árnu abecedu na pomocnej páske pre nejaké $\bar{k} \geq 1$. Keďže $g(n) = o(f(n))$, existuje dostatočne veľké $\bar{n}$ také, že $(m+m^2)g(\bar{n}) < f(\bar{n})$, kde $m = max\{log_2(j), log_2(\bar{k}+1), c\}$. Nech $\bar{d}$ je maximálne také prirodzené číslo, pre ktoré $(\bar{d}+\bar{d}^2)g(\bar{n}) < f(\bar{n})$. Zrejme $\bar{d} \geq m$.

Nech $\bar{v}$ je slovo dĺžky najviac $cg(\bar{n})$ nad touto $\bar{k}$-árnou abecedou, pomocou ktorého $\bar{A}$ akceptuje všetky slová z $L_{f,g}$ dĺžky $\bar{n}$. Keďže $L_{f,g}$ obsahuje jediné slovo dĺžky $\bar{n}$, označme ho $\bar{w}$, musí $\bar{A}$ akceptovať pomocou $\bar{v}$ jedine toto slovo $\bar{w}$ spomedzi slov dĺžky $\bar{n}$. Aby sme teraz dokázali, že dvojica $(\bar{A}, \bar{v})$ je jednoslovná pre vstupy dĺžky $\bar{n}$, stačí ešte dokázať, že $\bar{A}$ sa nachádza medzi prvými $2^{\bar{d}g(\bar{n})}$ automatmi v postupnosti S, že $\bar{A}$ má $\bar{k}$-árnu abecedu na pomocnej páske, kde $log_2(\bar{k}+1) \leq \bar{d}$ a že $\bar{v}$ je slovo dĺžky najviac $\bar{d}g(\bar{n})$, kde $\bar{d}$ je maximálne prirodzené číslo, pre ktoré $(\bar{d}+\bar{d}^2)g(\bar{n}) < f(\bar{n})$. Všetky tieto vlastnosti ale vyplývajú z toho, že $max\{log_2(j), log_2(\bar{k}+1), c\} \leq \bar{d}$ (pozri vyššie).

Teda dvojica $(\bar{A}, \bar{v})$ je jednoslovná pre vstupy dĺžky $\bar{n}$, pričom ale $\bar{A}$ akceptuje slovo $\bar{w} \in L_{f,g}$ dĺžky $\bar{n}$ s pomocou slova $\bar{v}$.

To je ale v spore s tým, ako boli volené slová do jazyka $L_{f,g}$ (pozri časť „druhý prípad“ vyššie), lebo v takom prípade slovo dĺžky n, ktoré bolo vybrané do jazyka

$L_{f,g}$, nemôže byť akceptované žiadnym automatom A' s pomocnou informáciou v', kde (A', v') je jednoslovná dvojica pre vstupy dĺžky n. Preto $L_{f,g} \notin NSPACE/g(n)$. □

Dôsledok 4.3.7. *Nech $f, g : \mathbb{N} \to \mathbb{N}$ je dvojica funkcií, pre ktoré platí, že $\forall n \in \mathbb{N} : 1 < f(n) \leq n, g(n) = o(f(n))$. Potom $DSPACE(1)/g(n) \subsetneq DSPACE(1)/f(n)$.*

Dôsledok 4.3.8. *Nech $f, g : \mathbb{N} \to \mathbb{N}$ je dvojica funkcií, pre ktoré platí, že $\forall n \in \mathbb{N} : 1 < f(n) \leq n, g(n) = o(f(n))$. Potom $NSPACE(1)/g(n) \subsetneq NSPACE(1)/f(n)$.*

4.4 Ohraničené jazyky

V tejto časti sa podrobnejšie pozrieme na ohraničené jazyky a na to, ako sú ich schopné akceptovať deterministické a nedeterministické konečné automaty s pomocnou páskou. Začneme definíciou ohraničeného jazyka.

Definícia 4.4.1. *Nech $w_1, \ldots, w_k, k \geq 1$ je konečná postupnosť neprázdnych slov nad abecedou Σ. Ohraničený jazyk L je potom ľubovoľná podmnožina jazyka $w_1^* \ldots w_k^*$.*

Veta 4.4.2. *Na akceptovanie ľubovoľného ohraničeného jazyka $L \subseteq w_1^* \ldots w_k^*$ pomocou $NFAT$ postačuje polynomiálna pomocná informácia.*

Dôkaz. To, že ohraničený jazyk obsahuje maximálne $(n+1)^k = O(n^k)$ slov dĺžky n vzhľadom na počet „zložiek" $w_1, \ldots, w_k$ nie je zložité ukázať. Horná hranica pre počet výskytov každej zo zložiek $w_1, \ldots, w_k$ je $n+1$ (keďže uvažujeme aj možnosť, že w_k sa v reťazci vôbec nevyskytne), inak by výsledné slovo bolo dlhšie. Keď toto ohraničenie aplikujeme na každú zo zložiek, dostávame $(n+1)^k$ rôznych slov jazyka L, pričom je zaručené, že v týchto slovách sú obsiahnuté všetky slová dĺžky presne n (a možno aj veľa iných, to nám ale neprekáža, lebo hľadáme horné ohraničenie).

To znamená, že tieto slová môžeme vypísať na pomocnú pásku a náš $NFAT$ si už len nedeterministicky tipne, pre ktoré zo slov má overiť rovnosť so vstupom. □

Veta 4.4.3. *Nech $\Sigma = \{a_1, \ldots, a_k\}$ je ľubovoľná abeceda. Na akceptovanie ohraničeného jazyka $L \subseteq a_1^* ... a_k^*$ nad Σ pomocou $DFAT$, pričom $\forall 1 \leq i \leq k : |a_i| = 1$, stačí pomocná informácia veľkosti $O(n^{k+1})$.*

Dôkaz. V dôkaze vety 4.4.2 sme odargumentovali, že ohraničené jazyky majú maximálne $(n+1)^k = O(n^k)$ slov dĺžky n, kde k je počet „zložiek" $w_1, \ldots, w_k$.

V pomocnej informácii pre vstupy dĺžky n budeme v špeciálnom formáte enumerovať slová jazyka L dĺžky n. Okrem symbolov $a_1, \ldots, a_k$ budeme používať ďalšie symboly $\bar{a_1}, \ldots, \bar{a_k}$. Nech L^n označuje množinu slov dĺžky n patriacich do L.

Pre každé slovo $z \in a_1^* a_2^* \dots a_k^*$, kde $0 \leq |z| \leq n-1$, definujme rekurzívne slovo $L(z)$. Nech $1 \leq |z| \leq n-1$ a nech najpravejší symbol slova z je a_j pre nejaké $1 \leq j \leq k$. Slovo $L(z)$ bude definované nasledovne:

1. Ak $|z| = n-1$, potom $L(z) = a'_k a'_{k-1} \dots a'_j$, kde $\forall l : a'_l = \begin{cases} \bar{a}_l, & \text{ak } za_l \in L^n \\ a_l, & \text{inak.} \end{cases}$
2. Ak $|z| \leq n-2$, potom $L(z) = a_k L(za_k) a_{k-1} L(za_{k-1}) \dots a_j L(za_j)$.
3. Ak $|z| = 0$, t.j. $z = \epsilon$, potom $L(\epsilon) = a_k L(a_k) a_{k-1} L(a_{k-1}) \dots a_1 L(a_1)$.

Lema 4.4.4. *Nech z je ľubovoľné slovo z $a_1^* \dots a_k^*$ dĺžky m, kde $1 \leq m \leq n-1$ a nech najpravejší symbol slova z je a_j pre nejaké $1 \leq j \leq k$. Potom slovo $L(z)$ neobsahuje žiadny symbol a_i ani $\bar{a}_i$ pre $i < j$.*

Dôkaz. Indukciou. Lema 4.4.4 zrejme platí pre slová dĺžky $n-1$, lebo v takom prípade $L(z) = a'_k a'_{k-1} \dots a'_j$. Predpokladajme teraz, že Lema 4.4.4 platí pre všetky slová dĺžky aspoň l pre nejaké $l \leq n-1$. Potom Lema 4.4.4 platí aj pre slová dĺžky $l-1$, lebo $L(z) = a_k L(za_k) a_{k-1} L(za_{k-1}) \dots a_j L(za_j)$ pre $|z| = l-1 \leq n-2$ a podľa indukčného predpokladu sa a_i ani $\bar{a}_i$ nemôže vyskytnúť v žiadnom slove $L(za_m)$ pre $m \geq j$ a symboly a_i, resp. $\bar{a}_i$ sú rôzne od symbolov $a_k, a_{k-1}, \dots, a_j$. □

Budeme hovoriť, že znak a_i je menší než znak a_j, ak $i < j$. Nech $w = w_1 \dots 2_n$ je ľubovoľné slovo dĺžky n, kde $w_l \in \{a_1, \dots, a_k\}$ pre všetky l. Je ľahké vidieť, že w nepatrí do $a_1^* \dots a_k^*$ a teda ani do $L \subseteq a_1^* \dots a_k^*$, ak symbol w_{l+1} je menší, než w_l pre nejaké l.

Nech A je $DFAT$, ktorý bude využívať uvedenú vlastnosť a ktorý bude nasledovne rozpoznávať slová dĺžky n patriace do L s pomocnou informáciou $L(\epsilon)$. Nech hlava na vstupnej páske číta znak $x \in \{a_1, \dots, a_k\}$. Automat najprv skontroluje, či znak x je menší, než znak na vstupe bezprostredne vľavo od x. Ak áno, potom A vstup zamietne. Ak nie, tak hlava automatu na pomocnej páske pôjde doprava, kým nenájde znak x (prípadne $\bar{x}$). Keď ho nájde, presunie sa o jeden znak doprava na vstupnej páske, nech je tam znak $y \in \{a_1, \dots, a_k\}$, a tiež sa posunie o jeden znak doprava na pomocnej páske.

Potom A opäť skontroluje, či znak y je menší než znak x. Ak áno, potom vstup zamietne. Ak nie, potom A hľadá znak y (prípadne $\bar{y}$) na pomocnej páske. Takto automat postupne príde až na posledný znak vstupného slova a pozrie sa na pomocnej páske, či nájdený znak je s pruhom. Ak áno, tak sa na vstupe nachádza slovo z L a teda A ho akceptuje, inak ho A zamietne.

Všimnime si, že pri výpočte automatu A na vstupe dĺžky n platí nasledujúci invariant.

Lema 4.4.5. *Nech za_m je ľubovoľný prefix ľubovoľného vstupu w dĺžky n, kde $z \in a_1^* \dots a_k^*$. Potom existuje krok výpočtu A na w, v ktorom hlava na vstupnej páske vstúpi na prvý symbol slova $L(z)$.*

Dôkaz. Indukciou. Najprv dokážeme tvrdenie pre základný prípad, t.j. $z = \epsilon$. V takomto prípade lema platí, lebo na začiatku výpočtu hlava na vstupe číta prvý symbol vstupu, t.j. číta a_m a hlava na pomocnej páske číta prvý symbol slova $L(\epsilon)$.

Predpokladajme teraz, že lema platí pre prefix za_m dĺžky i vstupu w. Dokážme, že lema platí aj pre prefix $za_m a_{m'}$ dĺžky $i+1$, kde $za_m \in a_1^* \dots a_k^*$. Podľa indukčného predpokladu existuje krok výpočtu A na w, v ktorom hlava na vstupnej páske číta najpravejší symbol prefixu za_m a v ktorom hlava na pomocnej páske číta prvý symbol slova $L(z) = a_k L(za_k) \dots a_j L(za_j)$, kde a_j je najpravejší symbol slova z.

Podľa algoritmu pre A (pozri vyššie), automat v ďalších krokoch porovná symboly a_m a a_j a zistí, že $m \geq j$, lebo podľa indukčného predpokladu platí $za_m \in a_1^* \dots a_k^*$. Preto A pokračuje vo výpočte, pričom hlavu na pomocnej páske presúva vpravo a hľadá najľavejší výskyt symbolu a_m (prípadne $\bar{a}_m$) v slove $L(z)$. Podľa lemy 4.4.4 je to výskyt symbolu a_m bezprostredne vľavo od podslova $L(za_m)$. Po nájdení tohto výskytu symbolu a_m sa obe hlavy posunú o jeden znak vpravo, čiže hlava na vstupej páske bude čítať najpravejší symbol prefixu $za_m a_{m'}$ a hlava na pomocnej páske bude čítať prvý symbol slova $L(za_m)$. □

Dokážme teraz, že A korektne rozpoznáva slová dĺžky n z jazyka L pomocou informácie $L(\epsilon)$. Uvažujme dva prípady.

V prvom prípade je na vstupe slovo $w = va_m$, kde $|w| = n$ a $w \in a_1^* \dots a_k^*$. Z lemy 4.4.5 priamo vyplýva, že keď hlava automatu na vstupnej páske vstúpi na posledný znak a_m vstupu w, vstúpi hlava na pomocnej páske na prvý symbol slova $L(v)$. Potom A porovná symboly a_m a a_j, kde a_j je najpravejší symbol slova v a A zistí, že $m \geq j$, lebo podľa predpokladu $w \in a_1^* \dots a_k^*$. Keďže $L(x) = a'_k \dots a'_j$ a $m \geq j$, automat A posunom hlavy na pomocnej páske nájde symbol a'_m a podľa toho, či $a'_m = \bar{a}_m$ alebo $a'_m = a_m$, A akceptuje alebo zamietne w.

V druhom prípade je na vstupe slovo w dĺžky n, kde $w \in a_1^* \dots a_k^*$ a teda $w \in L$. Nech z je prefix slova w maximálnej dĺžky, ktorý patrí do $a_1^* \dots a_k^*$. Nech za_l je prefix slova w dĺžky $|z|+1$. Z lemy 4.4.5 vyplýva, že keď hlava automatu na vstupnej páske vstúpi na najpravejší znak prefixu za_l, hlava na pomocnej páske vstúpi na prvý symbol slova $L(z)$. Potom A porovná symboly a_l a a_i, kde a_i je najpravejší symbol slova z a A zistí, že $l < i$, lebo prefix za_l nepatrí do $a_1^* \dots a_k^*$, keďže z je prefix maximálnej dĺžky patriaci do $a_1^* \dots a_k^*$. V takomto prípade A zamietne w.

Na záver odhadnime zhora dĺžku pomocnej informácie $L(\epsilon)$ pre vstupy dĺžky n.

Zo štruktúry slov $L(z)$ vyplýva, že sú dve možnosti, kde sa môžu jednotlivé symboly nachádzať v slove $L(\epsilon)$ vzhľadom na štruktúru slov $L(z)$.

Prvá možnosť je, že sa symbol nachádza bezprostredne vľavo od slova $L(z)$ pre nejaké slovo $z \in a_1^* \dots a_k^*$, kde $1 \leq |z| \leq n-1$. Takto umiestnené symboly sa nachádzajú v nanajvýš $n^{k+1]}$ pozíciach slova $L(\epsilon)$, lebo takýchto slov z je najviac $(n-1) \cdot n^k \leq n^{k+1}$, keďže slov $u \in a_1^* \dots a_k^*$ dĺžky $n-1$ je najviac n^k a každé slovo $z \in a_1^* \dots a_k^*$ dĺžky l, $1 \leq l \leq n-1$, je prefix dĺžky l niektorého z týchto slov u.

Druhá možnosť je, že sa symbol nachádza niekde v slove $L(z) = a'_k \dots a'_j$ pre nejaké $z \in a_1^* \dots a_k^*$, kde $|z| = n-1$ a kde a_j je najpravejší symbol slova z. takto umiestnené symboly sa nachádzajú v najviac $k \cdot n^l$ pozíciách slova $L(\epsilon)$, lebo $|L(z)| \leq k$ pre takéto slová z a počet takýchto slov z dĺžky $n-1$ je najviac n^k.

Z oboch možností vyplýva, že dĺžka pomocnej informácie $L(\epsilon)$ je najviac $n^{k+1} + k \cdot n^k = O(n^{k+1})$.

□

4.5 Ďalšie problémy

V tejto časti sa pozrieme na viacero zaujímavých problémov súvisiacich s konečnými automatmi s pomocnou informáciou a ich riešenia, čím ukážeme niektoré techniky na rozhodnutie o tom, či nedeterministické, prípadne aj deterministické konečné automaty s pomocnou informáciou dokážu daný jazyk akceptovať alebo nie.

4.5.1 Jazyky ostrej a neostrej nerovnosť binárnych reťazcov

Prvý takýto problém je rozpoznávanie jazyka $NUM_{\leq} = \{x\#y \mid x, y \in \{0,1\}^n, n \geq 0,$ *binárne číslo y nie je menšie než binárne číslo x*$\}$.

Chceme o ňom rozhodnúť, či sa dá akceptovať na jednosmerných nedeterministických konečných automatoch s polynomiálnou pomocnou informáciou. Tento jazyk nie je bezkontextový, čo sa dá ukázať pomocou pumpovacej lemy pre bezkontextové jazyky.

Jazyk $\mathbf{NUM_{\leq}}$

Veta 4.5.1 ([HMU01])**.** *(pumpovacia lema pre bezkontextové jazyky) K ľubovoľnému bezkontextovému jazyku L existujú čísla p, q také, že pre každé $w \in L$ také, že $|w| > p$ existujú u, v, x, y, z také, že*

1. $w = uvxyz$
2. $|vxy| \leq q$

3. $|vy| \geq 1$

4. $\forall i \geq 0 : uv^i xy^i z \in L$

Aby sme dokázali, že jazyk $NUM_{\leq}$ nie je bezkontextový, aplikujeme vetu 4.5.1 na slovo $w = 0^{p+2q}1^{2q}\#0^{2q}1^{p+2q}$. Ak sa pokúsime slovo w rozdeliť na vxy, zistíme, že prípustné rozdelenia sú len také, kde slovo x obsahuje mriežku a slová v a y sú rovnako dlhé (inak by pri „pumpovaní" jedna časť bola dlhšia ako druhá), pričom v nemôže obsahovať 0 a y nemôže obsahovať 1. Po dostatočnom počte „pumpovaní" teda dostaneme na ľavej strane číslo väčšie ako číslo na pravej strane a takéto slovo do jazyka $NUM_{\leq}$ už nepatrí.

Zároveň je zrejmé, že tento jazyk sa dá rozpoznať na lineárne ohraničených automatoch.

O tomto jazyku teraz sporom ukážeme, že sa nedá akceptovať pomocou $NFAT$ s polynomiálnou pomocnou informáciou.

Veta 4.5.2. $NUM_{\leq} \notin$ ***NSPACE(1)****/poly*

Dôkaz. Nech by existoval $NFAT$ A s polynomiálnou pomocnou informáciou akceptujúci jazyk $NUM_{\leq}$. Uvažujme dvojicu rovnako dlhých slov $a\#a$, $b\#b$ patriacich do $NUM_{\leq}$, kde a, b sú dva rôzne binárne reťazce, kde A bude po prečítaní mriežky v niektorom akceptačnom výpočte na slove $a\#a$ v tej istej slabej konfigurácii, v akej bude A po prečítaní mriežky v niektorom akceptačnom výpočte na slove $b\#b$. Z Dirichletovho princípu dostávame (podobne ako v dôkaze vety 4.2.4), že takáto dvojica slov musí existovať, lebo binárnych reťazcov danej dĺžky je exponenciálne veľa, kým A má len polynomiálne veľa možných slabých konfigurácii na týchto slovách. Bez ujmy na všeobecnosti, nech $a < b$. Uvažujme slovo $b\#a$.

Toto slovo bude A akceptovať, keďže stačí napojiť na seba vhodné časti z akceptačných výpočtov na $a\#a$ resp. $b\#b$. Zároveň z voľby a, b a z toho, že $b < a$, vyplýva, že slovo $b\#a$ nemôže patriť do $NUM_{\leq}$, čo je spor. □

Jazyk NUM$_<$

Uvažujme teraz namiesto neostrej nerovnosti ostrú. O tomto jazyku sa dá podobne ukázať, že nie je bezkontextový, ale akceptujú ho lineárne ohraničené automaty. Ukážeme, že ani tento jazyk sa nedá akceptovať na $NFAT$ s polynomiálnou pomocnou informáciou.

Veta 4.5.3. $NUM_{<} \notin$ ***NSPACE(1)****/poly*

Dôkaz. $+1$ budeme chápať ako aritmetickú operáciu - pripočítanie 1. Uvažujme dvojicu slov z jazyka $NUM_<$ tvaru $a\#a+1$, $b\#b+1$, kde a, b sú dva rôzne binárne reťazce také, že A sa v niektorom akceptačnom výpočte na slove $a\#a+1$ po prečítaní mriežky bude nachádzať v rovnakej slabej konfigurácii, v akej sa bude nachádzať v niektorom akceptačnom výpočte na slove $b\#b+1$ po prečítaní mriežky. Takáto dvojica reťazcov musí existovať, lebo počet takýchto slov danej dĺžky je exponenciálny, kým automat A má polynomiálny počet možných slabých konfigurácii. Bez ujmy na všeobecnosti, nech $b < a$. Uvažujme slovo $a\#b+1$. Vidno, že môžeme napojiť akceptačný výpočet nášho automatu A na slove $a\#a+1$ po mriežku na akceptačný výpočet automatu A na slove $b\#b+1$ od mriežky až po koniec, čím dostaneme akceptačný výpočet automatu A na slove $a\#b+1$, ktoré ale do jazyka $NUM_<$ zjavne nepatrí, lebo $b < a$, čo je spor. □

Záver

V práci sme poskytli zhrnutie známych výsledkov týkajúcich sa výpočtov konečných automatov s pomocnou informáciou. Takisto sme priniesli nové výsledky pre konečné automaty s pomocnou informáciou na dodatočnej páske. Ukázali sme, že pre jednosmerné $DFAT$ má zmysel uvažovať pomocnú informáciu dĺžky nanajvýš $O(n2^n)$, čo môže poskytnúť lepší náhľad pri riešení otvoreného problému, či exponenciálna pomocná informácia stačí na akceptovanie ľubovoľného jazyka na jednosmerných $DFAT$ a s tým súvisiaci problém, či vôbec PAL (jazyk palindrómov) sa dá akceptovať na jednosmerných $DFAT$ s exponenciálnou pomocnou informáciou.

Ďalším prínosom bolo, že sme porovnali silu deterministických a nedeterministických konečných automatov s pomocnou informáciou na dodatočnej stope, napr. na jazyku PAL, o ktorom sa ukázalo, že na $DFAT$ nestačí polynomiálna pomocná informácia a na $NFAT$ síce tiež nestačí, ale na komplement áno (vety 4.2.4 a 4.2.5). Podarilo sa nám takisto nájsť hierarchiu jazykov akceptovaných $DFAT$ s pomocnou informáciou sublineárnej dĺžky (veta 4.3.6), čím sme rozšírili výsledok o polynomiálnej hierarchii známy z [KSY13](veta 6) a ustanovili sme relatívne „hustú" hierarchiu jazykov akceptovaných $NFAT$.

Síce sa nám nepodarilo nájsť odpoveď na otvorený problém nastolený v [KSY13], ale veríme, že výsledky dosiahnuté v tejto práci poskytujú hlbší náhľad do problematiky a vidíme potenciál v ďalšom výskume nedeterminizmu v oblasti konečných automatov s pomocnou informáciou.

Literatúra

[DH95] Carsten Damm and Markus Holzer. Automata that take advice. In *Mathematical Foundations of Computer Science 1995, 20th International Symposium, MFCS'95, Prague, Czech Republic, August 28 - September 1, 1995, Proceedings*, pages 149–158, 1995. URL: http://dx.doi.org/10.1007/3-540-60246-1$_1$21.

[HMU01] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Longman Publishing, second edition, 2001.

[Kob85] Kojiro Kobayashi. Eleventh international colloquium on automata, languages and programming on the structure of one-tape nondeterministic turing machine time hierarchy. *Theoretical Computer Science*, 40:175 – 193, 1985. URL: http://www.sciencedirect.com/science/article/pii/0304397585901653.

[KSY13] Ugur Küçük, A. C. Cem Say, and Abuzer Yakaryilmaz. Finite automata with advice tapes. *CoRR*, abs/1312.2268, 2013. URL: http://dblp.uni-trier.de/db/journals/corr/corr1312.htmlKucukSY13.

[RG74] B.L. Rothschild R.L. Graham. A short proof of van der waerden's theorem on arithmetic progressions. *Proc. Amer. Math. Soc.*, 42:385–386, 1974.

[RS59] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM J. Res. Dev.*, 3(2):114–125, April 1959. URL: http://dx.doi.org/10.1147/rd.32.0114, http://dx.doi.org/10.1147/rd.32.0114 doi:10.1147/rd.32.0114.

[TYL03] Kohtaro Tadaki, Tomoyuki Yamakami, and Jack C. H. Lin. Theory of one tape linear time turing machines. *CoRR*, cs.CC/0310046, 2003. URL: http://arxiv.org/abs/cs.CC/0310046.

[Yam08] Tomoyuki Yamakami. Swapping lemmas for regular and context-free languages with advice. *CoRR*, abs/0808.4122, 2008. URL: http://arxiv.org/abs/0808.4122.

[Yam10] Tomoyuki Yamakami. The roles of advice to one-tape linear-time turing machines and finite automata. *CoRR*, abs/1007.3021, 2010. URL: http://arxiv.org/abs/1007.3021.