

Obmedzené použitie synchronizačných symbolov pri alternovaní

DIPLOMOVÁ PRÁCA

Marián Dudek

**UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA INFORMATIKY**

Informatika

Vedúci diplomovej práce
Prof. RNDr. Branislav Rován, PhD.

BRATISLAVA 2008

Čestné prehlásenie

Čestne prehlasujem, že túto diplomovú prácu som
vypracoval samostatne, len s použitím uvedenej literatúry.

Bratislava, 2008

Pod'akovanie

Rád by som na tomto mieste vyjadril vďaku svojmu diplomovému vedúcemu Prof. Rovanovi, za odborné konzultácie, ako aj za cenné rady a pripomienky pri písaní tejto práce. Taktiež by som rád poďakoval mojej rodine a priateľke Magdaléne Kovárovej, za ich podporu a pomoc počas celého môjho štúdia.

Abstrakt

Autor	Marián Dudek
Názov	Obmedzené použitie synchronizačných symbolov pri alternovaní
Škola	Univerzita Komenského v Bratislave
Fakulta	Fakulta matematiky, fyziky a informatiky
Katedra	Katedra informatiky
Školiteľ	Prof. RNDr. Branislav Rován, PhD.
Miesto	Bratislava
Rok	2008
Rozsah	50 strán
Výťah	V práci skúmame vplyv ohraničeného použitia synchronizačných symbolov pri alternovaní. Podrobnejšie skúmame model synchronizovane alternujúcich konečných automatov, ktoré synchronizujú s jedným symbolom a na záver výpočtu môžu synchronizovať s druhým symbolom ako "zarážkou".
Kľúčové slová	synchronizované alternovanie, alternovanie s jedným symbolom a zarážkou, označené zretazenie

Predhovor

Práca je z oblasti teoretickej informatiky, konkrétne Teórie formálnych jazykov a automatov. Vzniká v období, kedy sa vo veľkej miere vytvárajú softvérové riešenia, ktoré používajú viacero procesov, ktoré medzi sebou komunikujú. V našej práci sa zaoberáme modelom, ktorý túto situáciu popisuje v matematickom formalizme.

Hlavným cieľom je preskúmať, aké obmedzenia použitia synchronizačných symbolov majú vplyv na výpočtovú silu alternujúcich strojov a aké nie. Definovali sme niektoré obmedzenia a ukázali sme, akú výpočtovú silu má alternujúce zariadenie s konkrétnym druhom synchronizácie. Ak to bolo možné, hovorili sme o alternujúcich zariadeniach vo všeobecnosti – bez udania sekvenčného modelu, z ktorého vychádzajú. Pri obmedzení, v ktorom takýto prístup nevedol k výsledku, sme sa snažili ukázať výpočtovú silu modelu, ktorý vychádza z konkrétneho sekvenčného modelu.

Veríme, že výsledky, ku ktorým sme dospeli, budú viesť k lepšiemu pochopeniu toho, ako a prečo koncept Synchronizovaného alternovania pridáva sekvenčným automatom takú veľkú výpočtovú silu.

Obsah

1	Úvod	7
2	Definície a označenia	9
3	Synchronizovane alternujúce automaty s počítadlom	13
4	Počet synchronizačných symbolov	22
4.1	Alternovanie s jedným synchronizačným symbolom	22
4.2	Alternovanie s dvomi synchronizačnými symbolmi	25
5	Alternujúce konečné automaty s jedným synchronizačným symbolom a zarážkou	27
5.1	Porovnanie s Chomského hierarchiou	27
5.2	Rovnako synchronizujúci normálny tvar	35
5.3	Uzavretosť na označené zreťazenie	40
6	Záver	48
	Zoznam obrázkov	49
	Referencie	50

1 Úvod

Koncept Alternovania vznikol začiatkom osemdesiatych rokov, ako prirodzené rozšírenie nedeterminizmu. Bol publikovaný v [1]. Umožňuje modelovať výpočet, ktorý pozostáva z viacerých nezávislých procesov.

Na konci osemdesiatych rokov vznikol na Katedre informatiky koncept Synchronizovaného alternovania. Umožňuje modelovať výpočet, v ktorom jednotlivé dielčie procesy môžu medzi sebou komunikovať. Komunikácia prebieha v takzvaných synchronizačných stavoch a sú do nej zapojené všetky procesy, ktoré ešte neprešli do akceptačného stavu. Synchronizačný stav je tvorený dvojicou – stav procesu a synchronizačný symbol. Stav procesu vyjadruje lokálnu informáciu procesu, zatiaľ čo synchronizačný symbol je globálna informácia o výpočte, ktorú sa dozvie každý proces zapojený v komunikácii.

Výsledok, ktorý preslávil tento koncept hovorí, že synchronizovane alternujúce konečné automaty dokážu simulovať lineárne ohraničený automat. Výsledok je prekvapivý, pretože samotné Alternovanie konečným automatom nepridalo žiadnu výpočtovú silu, zatiaľ čo synchronizované alternovanie im umožnilo preskočiť dve triedy zložitosti. Za všetky práce venujúce sa tomuto konceptu spomeňme [2] a [3].

Koncept synchronizovane alternujúcich zariadení je zaujímavý aj z praktického hľadiska, pretože matematicky popisuje situáciu, s ktorou sa riešitelia softvérových systémov stretávajú veľmi často – niekoľko nezávislých procesov (bez spoločných hodín), ktoré počas výpočtu spolu komunikujú. Motiváciou pre takéto riešenie už často nebýva urýchlenie výpočtu. V súčasnosti je hlavnou motiváciou pre riešenie, v ktorom je výpočet rozdelený na viaceré nezávislé celky, zjednodušenie jedného veľkého problému na viaceré menšie problémy, z ktorých vyriešenia získame riešenie pôvodného veľkého problému. Čiže klasická taktika "Rozdeľuj a panuj".

Motiváciou pre skúmanie vplyvu obmedzenej synchronizácie na výpočtovú silu alternujúcich zariadení je, že koncept Synchronizovaného alternovania pridáva sekvenčným modelom veľkú výpočtovú silu. Dokazuje to okrem už spomínaného výsledku aj výsledok, že synchronizovane alternujúci počítadlový automat je rovnako výpočtovo silný ako Turingov stroj. Tento výsledok je prezentovaný v kapitole 3 a patrí medzi jednoduchšie.

V kapitole 4 sa budeme zaoberať obmedzením počtu synchronizačných symbolov. Ukážeme, že synchronizovanie len s jedným synchronizačným symbolom alternujúcim zariadeniam nepridá žiadnu výpočtovú silu. Naproti tomu alternujúce zariadenie, ktoré môže synchronizovať len v dvoch synchronizačných symboloch, je rovnako výpočtovo silné ako zariadenie, ktoré môže

synchronizovať bez obmedzenia.

Z predchádzajúceho je zrejmé, že je veľký rozdiel medzi synchronizovaním s jedným a s dvomi synchronizačnými symbolmi. Preto sa budeme snažiť nájsť spôsob synchronizovania, ktorý by zariadeniu pridal výpočtovú silu, ktorá je menšia ako výpočtová sila neobmedzene synchronizujúceho zariadenia. Výsledkom nášho snaženia bude koncept nazvaný Synchronizácia s jedným synchronizačným symbolom a zarážkou. V tomto koncepte alternujúce zariadenie môže počas výpočtu synchronizovať len s jedným symbolom a na konci výpočtu môže synchronizovať s druhým symbolom – "zarážkou". V kapitole 5 sa budeme zaoberať konečnými automatmi s takto obmedzenou synchronizáciou. Táto časť práce je najzaujímavejšia a prináša najzaujímavejšie výsledky.

Je ľahké vidieť, že takýto synchronizovane alternujúci konečný automat dokáže akceptovať jazyky, na ktorých akceptáciu nám nestačia ani bezkontextové prostriedky. Napríklad jazyk $a^n b^n c^n$.

Pre skúmaný model existuje normálny tvar, v ktorom všetky vlákna synchronizujú aspoň k krát a majú rovnakú synchronizačnú postupnosť (končiacu zarážkou).

Najzaujímavejší výsledok práce hovorí, že trieda definovaná alternujúcimi konečnými automatmi synchronizujúcimi s jedným symbolom a zarážkou, ktoré pre konkrétne slovo vyprodukurujú vždy rovnakú synchronizačnú postupnosť, je uzatvorená na označené zretáženie. Budeme prezentovať techniku, ktorá nám umožní do jednej synchronizačnej postupnosti zakódovať dve synchronizačné postupnosti.

2 Definície a označenia

V tejto kapitole uvedieme niektoré definície, ktoré sú relevantné k tvrdeniam nachádzajúcim sa v tejto práci. Budeme používať obvyklú notáciu pre pojmy konfigurácia a krok výpočtu (\vdash_A) zariadenia. Ich formálne definície môžeme nájsť napríklad v [4].

Predtým, než uvedieme formálne definície synchronizovane alternujúcich zariadení, v krátkosti popíšeme ako vyzerá výpočtový proces takéhoto zariadenia. Na začiatku výpočtu je jeden proces, ktorý je v počiatočnom stave a na jednosmernej read only páske má vstupné slovo. Alternovanie je prirodzené rozšírenie nedeterminizmu. Nedeterminizmus umožňuje automatu z jednej konfigurácie prejsť do inej z viacerých možných. Vstupné slovo je akceptované, ak existuje výpočet, ktorý skončí v akceptačnej konfigurácii (je prečítaný celý vstup a automat je v akceptačnom stave). Pri alternovaní dovoľujeme aj prechody z jednej konfigurácie do viacerých konfigurácii naraz. Jeden rodičovský proces sa rozdelí na viaceré dcérske procesy. Dcérske procesy budú mať rovnakú pamäť a pozíciu na vstupnom slove ako rodičovský proces, až na prechod podľa δ funkcie. Slovo bude akceptované, ak existuje výpočet, v ktorom sa všetky procesy dostanú do akceptačnej konfigurácie.

Na synchronizáciu sa dá neformálne pozrieť nasledovne. Ak procesor prejde do synchronizačného stavu (ten predstavuje dvojicu stav a synchronizačný symbol), tak čaká, až všetky živé procesory prejdú do synchronizačného stavu. Výpočet môže zase pokračovať, ak všetky procesory mali rovnaký synchronizačný symbol. Inak sa výpočet zasekne. Rovnako ako pri alternovaní, slovo bude akceptované ak existuje výpočet, v ktorom všetky procesy dosiahnu akceptačnú konfiguráciu.

Definícia 1. *Synchronizovane alternujúci konečný automat je 7-mica $(Q, S, A, \Sigma, \delta, q_0, F)$, kde:*

- Q - Konečná množina stavov
- S - Konečná množina synchronizačných symbolov
- $Q \cup (Q \times S)$ - Interný stav automatu, budeme označovať Q'
- $A \subset Q'$ - Množina univerzálnych stavov
- Σ - Konečná vstupná abeceda
- $\Sigma \cup \{\varepsilon\}$ budeme označovať Σ_ε
- $\delta : Q' \times \Sigma_\varepsilon \rightarrow 2^{Q'}$ - Prechodová funkcia
- $q_0 \in Q$ - Počiatočný stav
- $F \subset Q$ - Množina akceptačných stavov

Zariadenie budeme označovať ako SAKA.

Definícia 2. *Synchronizovane alternujúci počítadlový automat je 8-ica $(Q, S, A, \Sigma, \delta, q_0, F)$, kde:*

Q, S, A, Σ, q_0, F - Rovnako ako pre SAKA
 x - Symbol počítadla
Množinu $\{\varepsilon, x\}$ bude označovať X_ε

$\delta : Q' \times \Sigma_\varepsilon \times X_\varepsilon \rightarrow 2^{Q' \times x_\varepsilon^*}$ - Prechodová funkcia

Zariadenie budeme označovať SAPA.

Hodnota uložená v počítadle bude určená počtom symbolov x (v zásobníku, ak počítadlový automat chápeme ako zásobníkový automat s jedno prvkovou pracovnou abecedou).

Budeme predpokladať model počítadlových automatov, v ktorom ak je počítadlo neprázdne, sa vždy prečíta symbol x , inak sa prečíta ε .

Automat s počítadlom bude v akceptačnej konfigurácii, ak jeho stav bude v množine akceptačných stavov.

Definícia 3. *Synchronizovane alternujúci konečný automat s jedným synchronizačným symbolom a zarážkou je 6-tica $(Q, A, \Sigma, \delta, q_0, F)$, kde:*

Q, A, Σ - Rovnako ako pre SAKA
 $Q \cup (Q \times \{1\})$ - Výpočtový interný stav automatu, budeme označovať Q_1
 $Q_1 \cup (F \times \{2\})$ - Interný stav automatu, budeme označovať Q_2
 $\delta : Q_1 \times \Sigma_\varepsilon \rightarrow 2^{Q_2}$ - Prechodová funkcia
 $q_0 \in Q$ - Počiatočný stav
 $F \subset Q$ - Množina F
 $F \cup (F \times \{2\})$ - Množina interných akceptačných stavov

Zariadenie budeme označovať ako S_2^1AKA . Všimnite si, že ak sa proces dostane do interného stavu s druhým synchronizačným symbolom a jeho konfigurácia nie je akceptačná, tak proces nebude akceptovať. Z tejto konfigurácie už neexistuje prechod.

Označenie 1. *Niekedy budeme hovoriť o synchronizovane alternujúcich zariadeniach vo všeobecnosti, bez udania konkrétneho sekvenčného modelu, z ktorého vychádzajú. Vtedy budeme používať označenie SAZ, prípadne S_2^1AZ .*

Označenie 2. *Interný synchronizačný stav budeme označovať ako dvojicu stav, symbol v hranatých zátvorkách, t.j. [\langle stav \rangle , \langle symbol \rangle].*

Definícia 4. *Budeme hovoriť, že konfigurácia je univerzálna, ak je jej stav z množiny A . Inak bude konfigurácia existenčná.*

Budeme hovoriť, že konfigurácia je synchronizačná, ak jej stav je z množiny $Q \times S$. Inak budeme hovoriť, že konfigurácia je nesynchronizačná.

Definícia 5. Úplný strom konfigurácii synchronizovane alternujúceho stroja A na vstupnom slove w je (možno aj nekonečný) strom s označenými vrcholmi T_w^A taký, že:

- Každý uzol t stromu T_w^A je označený konfiguráciou $c(t)$ stroja A
- Koreň t_0 je označený počiatočnou konfiguráciou stroja A na slove w
- t_2 je priamy potomok t_1 práve vtedy, keď $c(t_1) \vdash_A c(t_2)$

$c(v)$ označuje konfiguráciu vrcholu v .

$c_1 \vdash_A c_2$ označuje reláciu krok výpočtu pre zariadenie A .

Niekedy budeme používať označenie $sc(v)$ pre stav konfigurácie, ktorou je označený vrchol v .

Definícia 6. Synchronizačná postupnosť uzla t v úplnom výpočtovom strome T_w^A s koreňom t_0 je postupnosť synchronizačných symbolov vyskytujúcich sa na označeniach uzlov na ceste z t_0 do t .

Synchronizačnú postupnosť uzla t budeme označovať $SP(t)$.

Označenie 3. Symbolom $\mathbb{N}_{l,r}$ budeme označovať množinu $\{l, l+1, l+2, \dots, r\}$. Symbolom \mathbb{N}_r budeme označovať množinu $\mathbb{N}_{1,r}$.

Označenie 4. Budeme hovoriť, že postupnosť $a = a_1 a_2 \dots a_n$ je počiatočnou podpostupnosťou postupnosti $b = b_1 b_2 \dots b_m$ ak platí:

$$(n \leq m) \wedge (\forall i \in \mathbb{N}_n) (a_i = b_i)$$

Túto skutočnosť budeme označovať $a \prec b$.

Definícia 7. Výpočtový strom zariadenia A na vstupnom slove w je (možno aj nekonečný) podstrom T' úplného stromu konfigurácii T stroja A na w taký, že:

1. Každý uzol v T' označený univerzálnou konfiguráciou má rovnakých priamych nasledovníkov ako v T , alebo nemá žiadneho nasledovníka
2. Každý uzol v T' označený existenčnou konfiguráciou má najviac jedného priameho nasledovníka
3. Pre všetky uzly u, v z T' platí, že $SP(u) \prec SP(v)$ alebo $SP(v) \prec SP(u)$

Niekedy budeme hovoriť o výpočte zariadenia A na vstupnom slove w . Budeme tým chápať podstrom výpočtového stromu, ktorý obsahuje koreň.

Definícia 8. Akceptačný výpočet zariadenia A na vstupnom slove w je konečný výpočtový strom A na slove w taký, že každý list je označený akceptačnou konfiguráciou.

Definícia 9. Vlákno výpočtového stromu je orientovaná cesta z koreňa alebo priameho nasledovníka vrcholu označeného univerzálnou konfiguráciou do najbližšieho vrcholu označeného univerzálnou alebo akceptačnou konfiguráciou.

Definícia 10. Nech A je synchronizovane alternujúce zariadenie. Budeme hovoriť, že A je $T(n)$ časovo ohraničený, ak pre každé slovo $w \in L(A)$ dĺžky n existuje akceptačný výpočet, ktorého hĺbka je navyše $T(n)$.

Definícia 11. Budeme hovoriť, že synchronizovane alternujúce zariadenie A je deterministicky synchronizujúce, ak pre každé vstupné slovo w a ľubovoľné akceptačné výpočty T_1, T_2 zariadenia A na slove w platí, že ak vrchol v má najdlhšiu synchronizačnú postupnosť z vrcholov T_1 a u má najdlhšiu synchronizačnú postupnosť z vrcholov T_2 , tak $SP(v) = SP(u)$.

Definícia 12. Počítadlový automat s dvomi počítadlami je 5-ica $(Q, \Sigma, \delta, q_0, F)$.

Q, Σ, q_0, F - Ako pri SAKA
 $\delta : Q \times \Sigma_\varepsilon \times X_\varepsilon \times X_\varepsilon \rightarrow 2^{Q \times X_\varepsilon \times X_\varepsilon}$ - Prechodová funkcia
 Zariadenie budeme označovať P2A.

3 Synchronizovane alternujúce automaty s počítadlom

V tejto kapitole sa budeme zaoberať výpočtovou silou synchronizovane alternujúcich zariadení, ktorých model vychádza z modelu sekvenčných automatov s počítadlom. Ukážeme, že je rovnako výpočtovo silný ako Turingov stroj. Simulácia Turingovho stroja nebude priama. Ukážeme, že synchronizovane alternujúci automat s počítadlom dokáže simulovať automat s dvomi počítadlami a ten dokáže simulovať Turingov stroj. Simuláciu Turingovho stroja automatom s dvomi počítadlami možno nájsť napr. v [5].

Príklad 3.1. *Nech M je počítadlový automat s dvomi počítadlami. Definujeme synchronizovane alternujúci počítadlový automat M' nasledovne:*

Nech M je $(Q, \Sigma, \delta, q_0, F)$. Automat M' bude fungovať nasledovne: Najskôr sa rozdelí na dva procesy. Proces A bude simulovať prvé počítadlo a jeho stav bude zodpovedať stavu simulovaného automatu M . Proces B bude simulovať druhé počítadlo. Každý krok simulácie bude vyzerať nasledovne:

Z pohľadu procesu A :

Proces A si tipne, aký symbol i je na vrchu druhého počítadla. Následne urobí prechod simulujúci automat M . Ak podľa δ_M treba vložiť do druhého zásobníka slovo j , proces bude synchronizovať v symbole (i, j) .

Ak je proces A v stave, ktorý zodpovedá akceptačnému stavu automatu M , tak môže synchronizovať v symbole ϵ a prejsť do akceptačného stavu. Tento technický krok nám umožňuje viac ohraničiť výpočet automatu M' a zjednoduší nám dôkaz jeho korektnosti. Umožní nám ukázať, že oba procesy skončia výpočet v rovnakom čase.

Z pohľadu procesu B :

Proces B bude počas výpočtu v novom stave q_b , resp. q'_b . Pri každom kroku výpočtu nedeterministicky uhádne, či proces A chce prejsť do akceptačného stavu. V pozitívnom prípade synchronizuje v symbole ϵ a prejde do akceptačného stavu. V negatívnom simuluje krok automatu M . Nech symbol i je na vrchu jeho počítadla. Proces si tipne aké slovo j sa má v simulovanom kroku výpočtu zapísať do druhého počítadla. Vloží j do svojho počítadla a synchronizuje v symbole (i, j) .

Proces B sa najskôr nachádza v stave q_b . V tomto stave číta pri každom kroku výpočtu symbol zo vstupného slova. Keď prečíta celé vstupné slovo, prejde do stavu q'_b , v ktorom už vstupné slovo nečíta. Aj toto je technické riešenie, ktoré ohraničuje výpočet automatu M' a zjednoduší nám dôkaz jeho korektnosti.

Nech funkcia f je definovaná nasledovne:

$$f : X_\varepsilon \rightarrow \{0, 1\}$$

$$f(z) = \begin{cases} 1 & z = x \\ 0 & \text{inak} \end{cases}$$

Aby sme sa vyhli zbytočným zátvorkám, budeme hodnotu $f(z)$ zapisovať ako f_z . Prirodzené rozšírenie funkcie f na reťazce budeme označovať g (napr. $g_{xxx} = 111$).

Nech M' je $(Q', S', A', \Sigma, \delta', q'_0, F')$, kde:

$$Q' = \{q'_0, q'_f, q_b, q'_b\} \cup Q$$

$$X = \{j \mid \exists q \in Q, k, i \in X_\varepsilon, a \in \Sigma_\varepsilon, (p, l, j) \in \delta(q, a, k, i)\}$$

$$S' = \{2\} \cup \{(i, j) \mid i \in X_\varepsilon, j \in X\}$$

$$A' = \{q'_0\}$$

$$F' = \{q'_f\}$$

Funkcia δ' je definovaná nasledovne:

$$\delta'(q'_0, \varepsilon, \varepsilon) = \{([q_0, (0, 0)], \varepsilon), ([q_b, (0, 0)], \varepsilon)\} \quad (3.1)$$

$$\delta'([q', (f_{i'}, g_{j'})], a, k') = \{([q, (f_i, g_j)], k) \mid i \in X_\varepsilon, (q, k, j) \in \delta(q', a, k', i), \\ q' \in Q, i' \in X_\varepsilon, j' \in X, a \in \Sigma_\varepsilon, k' \in X_\varepsilon\} \quad (3.2)$$

$$\delta'([q, (f_i, g_j)], \varepsilon, k) = \{([q'_f, 2], k)\}, \quad i, k \in X_\varepsilon, j \in X, q \in F \quad (3.3)$$

$$\delta'([q_b, (f_{i'}, g_{j'})], a, i) = \{([q_b, (f_i, g_j)], j) \mid j \in X\} \\ i, i' \in X_\varepsilon, j' \in X, a \in \Sigma \quad (3.4)$$

$$\delta'([q_b, (f_{i'}, g_{j'})], \varepsilon, i) = \{([q'_b, (f_i, g_j)], j) \mid j \in X\} \\ \cup \{([q'_f, 2], i)\} \cup \{([q'_b, (f_i, g_j)], j) \mid j \in X\} \\ i, i' \in X_\varepsilon, j' \in X \quad (3.5)$$

$$\delta'([q'_b, (f_{i'}, g_{j'})], \varepsilon, i) = \{([q'_b, (f_i, g_j)], j) \mid j \in X\} \cup \{([q'_f, 2], i)\}, \\ i, i' \in X_\varepsilon, j' \in X \quad (3.6)$$

$$\delta'([q'_f, 2], k) = \{(q'_f, k)\}, \quad k \in X_\varepsilon \quad (3.7)$$

Lema 3.1. *Nech T_w^M je akceptačný výpočet stroja M na vstupnom slove w . Nech konfigurácia, ktorá sa vyskytuje vo výpočte T_w^M ako n -tý člen je $([q, (f_i, g_j)], m, k, l)$. Členy akceptačného výpočtu číslujeme od 0, t.j. konfigurácia $(q_0, w, \varepsilon, \varepsilon)$ je 0-tý člen. Potom existuje výpočet $T_w^{M'}$ stroja M' na vstupnom slove w , ktorý obsahuje dve vlákna a vrcholy, ktorých cesta od koreňa obsahuje $n + 2$ vrcholov, ktoré majú rovnakú synchronizačnú postupnosť a sú označené konfiguráciami k_1, k_2 , kde:*

$$\begin{aligned} k_1 &= ([q, (f_i, g_j)], m, k) \\ k_2 &= ([b_n^w, (f_i, g_j)], w_n, l) \\ b_n^w &= \begin{cases} q_b & n \leq |w| \\ q'_b & \text{inak} \end{cases} \\ w_n &= \begin{cases} \text{suffix } w \text{ dlhý } |w| - n & n < |w| \\ \varepsilon & \text{inak} \end{cases} \end{aligned}$$

Dôkaz. Dôkaz urobíme matematickou indukciou vzhľadom na n .

1° $n = 1$

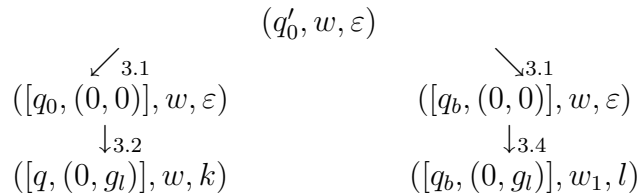
Relevantná časť výpočtu stroja M je

$$(q_0, w, \varepsilon, \varepsilon) \vdash_M (q, m, k, l)$$

a relevantný člen je (q, m, k, l) . Podľa definície funkcie δ' existuje výpočet, ktorý je znázornený na obrázku 3.1. Dolné indexy pri šípkach hovoria, podľa ktorej časti definície funkcie δ' bol uskutočnený prechod. Tento výpočet spĺňa vlastnosti:

- Obsahuje dve vlákna
- Vrcholy ktorých cesta od koreňa obsahuje tri vrcholy, majú rovnakú synchronizačnú postupnosť a to $(0, l)$ a sú označené konfiguráciami $([q, (0, g_l)], w, k), ([b_n^w, (0, g_l)], w_1, l)$.

Čiže tvrdenie pre $n = 1$ platí.



Obrázok 3.1: SAPA simulujúci P2A, prvý krok indukcie

2° Predpokladajme, že tvrdenie platí pre $n \geq 1$. Ukážme, že platí aj pre $n + 1$.

Nech n -tý a $(n + 1)$ -vý člen T_w^M sú konfigurácie $(q, m, k, l), (q', m', k', l')$.

Podľa indukčného predpokladu existuje výpočet stroja M' na slove w , ktorý obsahuje dve vlákna a vrcholy, ktorých cesta od koreňa obsahuje $n + 2$ vrcholov, ktoré sú označené konfiguráciami $([q, (f_i, g_j)], w, k), ([b_n^w, (f_i, g_j)], w_n, l)$.

Keďže existuje prechod $(q, m, k, l) \vdash_M (q', m', k', l')$, tak podľa 3.2 z definície funkcie δ' existuje prechod $([q, (f_i, g_j)], m, k) \vdash_{M'} ([q', (f_{i'}, g_{j'})], m', k')$, kde i', j' sú vhodné.

Rozoberme tri prípady:

1. Ak $n < |w|$, tak $b_n^w = q_b, b_{n+1}^w = q_b$ a podľa 3.4 z definície funkcie δ' existuje prechod

$$([q_b, (f_i, g_j)], w_n, l) \vdash_{M'} ([q_b, (f_{i'}, g_{j'})], w_{n+1}, l')$$

2. Ak $n = |w|$, tak $b_n^w = q_b, b_{n+1}^w = q'_b$ a podľa 3.5 z definície funkcie δ' existuje prechod

$$([q_b, (f_i, g_j)], w_n, l) \vdash_{M'} ([q'_b, (f_{i'}, g_{j'})], w_{n+1}, l')$$

3. Ak $n > |w|$, tak $b_n^w = q'_b, b_{n+1}^w = q'_b$ a podľa 3.6 z definície funkcie δ' existuje prechod

$$([q_b, (f_i, g_j)], w_n, l) \vdash_{M'} ([q'_b, (f_{i'}, g_{j'})], w_{n+1}, l')$$

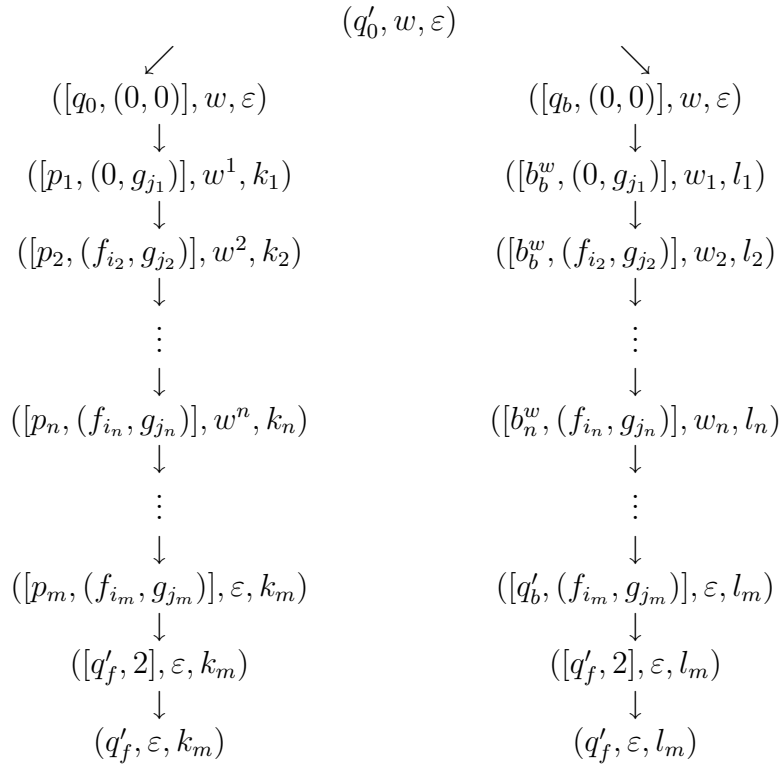
Z predchádzajúceho vyplýva, že existuje výpočet stroja M' na vstupnom slove w , ktorý spĺňa požadované vlastnosti. Výpočet je znázornený na obrázku 3.2. Čiže tvrdenie platí aj pre $n + 1$.

$$\begin{array}{ccc} \vdots & & \vdots \\ ([q, (f_i, g_j)], m, k) & & ([b_n^w, (f_i, g_j)], w_n, l) \\ \downarrow & & \downarrow \\ ([q', (f_{i'}, g_{j'})], m', k') & & ([b_{n+1}^w, (f_{i'}, g_{j'})], w_{n+1}, l') \end{array}$$

Obrázok 3.2: SAPA simulujúci P2A, indukčný krok

3° Z 1. a 2. kroku indukcie vyplýva, že tvrdenie platí pre ľubovoľné n \square

Lema 3.2. *Nech $T_w^{M'}$ je akceptačný výpočet stroja M' na ľubovoľnom vstupnom slove w . Potom výpočet $T_w^{M'}$ je tvaru podľa obrázku 3.3 a pre každý pár vrcholov v, u , ktorých cesta od koreňa obsahuje $n + 2$ vrcholov a ktorých konfigurácie sú $([q, (f_i, g_j)], m, k), ([b_n^w, (f_i, g_j)], w_n, l)$ platí, že $SP(v) = SP(u)$ a existuje výpočet T_w^M stroja M na slove w , ktorý obsahuje konfiguráciu (q, m, k, l) . Navyše platí, že $p_m \in q_f$, i_o je prvý znak z l_{o-1} a postupnosť l_o sa rovná postupnosti l_{o-1} bez prvého znaku a spredu zväčšená o slovo j_o*



Obrázok 3.3: Tvar výpočtu SAPA M' na slove

Dôkaz. Najskôr dokážme tvar pre prvé dve úrovne výpočtu a že $T_w^{M'}$ používa dve vlákna.

1. Konfigurácia $c_0 = (q'_0, w, \varepsilon)$ nie je akceptačná $\rightsquigarrow \forall T_w^{M'}$ existuje prechod z konfigurácie c_0
2. Definícia funkcie δ' \rightsquigarrow Existuje jediný prechod z konfigurácie c_0 do konfigurácií $c_1^1 = ([q_0, (0, 0)], w, \varepsilon), c_1^2 = ([q_b, (0, 0)], w, \varepsilon)$
3. (1.) \wedge (2.) \rightsquigarrow Prvé dve úrovne $T_w^{M'}$ odpovedajú obrázku 3.3

4. (3.) $\wedge A = \{q'_0\} \wedge$ Do stavu q'_0 sa nedá vrátiť $\rightsquigarrow T_w^{M'}$ používa dve vlákna

Teraz dokážme tvar výpočtu pre posledné tri úrovne:

1. $F' = \{q'_f\} \rightsquigarrow$ Posledná úroveň zodpovedá obrázku 3.3
2. (1.) \wedge Do stavu q'_f sa dá dostať len zo stavu $[q'_f, 2]$. Pri prechode nebude prečítané žiadne písmeno zo vstupného slova a obsah počítadla sa nezmení \rightsquigarrow Konfigurácie poslednej úrovne zodpovedajú obrázku 3.3
3. Je ľahké vidieť, že pred synchronizáciou v symbole 2 sa v prvom vlákne používajú stavy tvaru $[p, (f_i, g_j)]$ a v druhom vlákne stavy tvaru $[q_b, (f_i, g_j)]$, resp. $[q'_b, (f_i, g_j)]$
4. (2.) \wedge (3.) \wedge Prechod do stavu $[q'_f, 2]$ je možný len zo stavov $[q, (i, j)]$, $[q-b, (i', j')]$, kde $q \in F$. Pri tomto prechode nebude prečítané žiadne písmeno zo vstupného slova a obsah zásobníka sa nezmení \rightsquigarrow Konfigurácie predpredposlednej úrovne zodpovedajú obrázku 3.3

Ostatné úrovne tvaru výpočtu, ich napojenie na úrovne, ktorých tvar sme už ukázali a existenciu príslušného člena vo výpočte stroja M na slove w dokážeme indukciou vzhľadom na n .

1° $n = 1$

Nech c_2^1 je konfigurácia, do ktorej sa stroj M' dostal z konfigurácie c_1^1 . Podľa definície δ' mohli nastať nasledovné možnosti:

1. $c_2^1 = ([p_1, (f_{i_1}, g_{j_1})], w^1, k_1) \rightsquigarrow$ Bolo použité pravidlo 3.2 a teda z počiatočnej konfigurácie stroja M na slove w rovnvej $(q_0, w, \varepsilon, \varepsilon)$ sa dá dostať do konfigurácie (p_1, w^1, k_1, j_1) .
2. $c_2^1 = ([q'_f, 2], w, k_1) \wedge$ Definícia funkcie $\delta' \wedge$ Predošlí odsek \wedge Zo stavu q'_f nie je možný žiadny prechod \rightsquigarrow Ide o predpredposlednú úroveň

Nech c_2^2 je konfigurácia, do ktorej sa stroj M' dostal z konfigurácie c_1^2 . Synchronizačný symbol v konfigurácii c_2^2 musí byť rovnaký ako pri c_2^1 , lebo inak by sa výpočet zasekol. Teraz ukážme, že proces B musí zodpovedať obrázku 3.3

1. c_2^1 obsahuje synchronizačný symbol $(f_{i_1}, g_{j_1}) \wedge w = \varepsilon \wedge$ Definícia funkcie $\delta' \rightsquigarrow$ Mohol byť použitý len prechod do stavu, ktorého prvá zložka je $q'_b = b_n^w$

2. Definícia funkcie $\delta' \rightsquigarrow$ Zo stavu q'_b sa nedá dostať do stavu, v ktorom by sa čítalo vstupné slovo
3. (2.) $\wedge c_2^1$ obsahuje synchronizačný symbol $(f_{i_1}, g_{j_1}) \wedge w \neq \varepsilon \wedge$ Definícia funkcie $\delta' \rightsquigarrow$ Mohol byť použitý len prechod do stavu, ktorého prvá zložka je $q_b = b_n^w$
4. c_2^1 obsahuje synchronizačný symbol 2 \rightsquigarrow Mohol byť použitý len prechod do stavu, ktorého prvá zložka je q'_f a teda ide o predpredposlednú úroveň

Vo všetkých prípadoch stav procesov zodpovedá obrázku 3.3 a v prípade, že stavy procesov sú $([p_1, (0, g_{j_1})], w^1, k_1), ([b_n^w, (0, g_{j_1})], w_n, l_1)$, existuje výpočet stroja M na slove w , ktorý obsahuje konfiguráciu (p_1, w^1, k_1, l_1) . Čiže tvrdenie pre $n = 1$ platí.

2° Predpokladajme, že tvrdenie platí pre $n \geq 1$. Ukážme, že tvrdenie platí aj pre $n + 1$

1. $IP \rightsquigarrow$ Existujú vrcholy s rovnakou synchronizačnou postupnosťou, ktoré sú označené konfiguráciami $c_n^1 = ([p_n, (f_{i_n}, g_{j_n})], w^n, k_n)$, $c_n^2 = ([b_n^w, (f_{i_n}, g_{j_n})], w_n, l_n)$ a existuje výpočet stroja M na slove w , ktorý obsahuje člen (p_n, w^n, k_n, l_n) .
2. Nech c_{n+1}^1, c_{n+1}^2 sú konfigurácie, do ktorých sa automat M' dostal s konfigurácií c_n^1, c_n^2 .
3. Zo stavov s prvou zložkou q'_b resp. q'_f sa nedá dostať do stavu, v ktorom by sa čítalo vstupné slovo
4. (3.) $\wedge b_n^w = q_b \wedge w_n \neq \varepsilon \rightsquigarrow c_{n+1}^2$ má prvú zložku stavu $q_b = b_{n+1}^w \wedge$ Zvyšok vstupného slova je v c_{n+1}^2 je o prvý znak kratší ako w_n , čiže je rovný w_{n+1}
5. $b_n^w = q_b \wedge w_n = \varepsilon \wedge$ Predkovia vrcholu označeného konfiguráciou c_{n+1}^2 majú vždy o jeden znak kratší zvyšok vstupného slova $\wedge c_{n+1}^2$ nesynchronizuje v symbole 2 $\rightsquigarrow c_{n+1}^2$ má prvú zložku stavu rovnú $q'_b = b_{n+1}^w$ alebo $q'_f \wedge$ Zvyšok vstupného slova je $\varepsilon = w_{n+1}$
6. (3.) $\wedge b_n^w = q'_b \wedge c_{n+1}^2$ nesynchronizuje v synchronizačnom symbole 2 $\rightsquigarrow q'_b = b_{n+1}^w \wedge$ Zvyšok vstupného slova je w_{n+1}
7. (1.) \wedge (4.) \wedge (5.) \wedge (6.) $\wedge c_{n+1}^2$ nesynchronizuje v symbole 2 \wedge Definícia funkcie $\delta' \rightsquigarrow c_{n+1}^2 = ([b_{n+1}^w, (f_{i_{n+1}}, g_{j_{n+1}})], w_{n+1}, l_{n+1})$, kde i_{n+1} je prvý znak z l_{n+1}

8. (1.) \wedge (7.) \wedge Definícia funkcie $\delta' \wedge c_{n+1}^2$ nesynchronizuje v symbole 2 $\rightsquigarrow c_{n+1}^1 = ([p_{n+1}, (f_{i_{n+1}}, g_{j_{n+1}})], w^{n+1}, k_{n+1}) \wedge$ Existuje výpočet stroja M na slove w , ktorý je rovnaký ako výpočet z (1.) a po (p_n, w^n, k_n, l_n) nasleduje člen $(p_{n+1}, w^{n+1}, k_{n+1}, l_{n+1})$
9. c_{n+1}^2 synchronizuje v symbole 2 \wedge Zo stavu $[q'_f, 2]$ sa dá dostať len do stavu q'_f a z tohto stavu už nie je možný prechod \wedge Predchádzajúce dokázané vlastnosti o tvare výpočtu \rightsquigarrow Ide o predpredposlednú úroveň
10. (8.) \wedge (9.) \rightsquigarrow Tvrdenie platí aj pre $n + 1$

3° Z prvého a druhého kroku vyplýva platnosť tvrdenia pre ľubovoľné n □

Veta 3.1. *Nech M je počítadlový automat s dvomi počítadlami. Potom existuje synchronizovane alternujúci počítadlový automat M' , ktorý akceptuje rovnaký jazyk. Navyiac ak M je $T(n)$ časovo ohraničený, tak M' je $T(n) + 2$ časovo ohraničený.*

Dôkaz. $L(M) \subset L(M')$

1. Nech w je slovo z jazyka $L(M)$ \rightsquigarrow Existuje akceptačný výpočet T_w^M stroja M na slove w . Nech tento výpočet má n -členov
2. (1.) \wedge Definícia akceptačného výpočtu \rightsquigarrow Posledný člen T_w^M je označený konfiguráciou (q, ε, k, l) , kde $q \in F$.
3. (2.) \wedge Lema 3.1 \rightsquigarrow Existuje výpočet stroja M' na slove w , ktorý obsahuje dve vlákna. Na prvom vlákne sa vyskytuje vrchol v a na druhom vrchol u také, že:
$$SP(u) = SP(v)c(v) = ([q, (f_i, g_j)], \varepsilon, k), \quad q \in Fc(u) = ([b_n^w, (f_i, g_j)], w_n, l)$$
4. (3.) \wedge Na prvom vlákne je zvyšok vstupného slova rovný ε $\rightsquigarrow n \geq |w|$
5. (3.) \wedge (4.) $\rightsquigarrow c(u) = ([q'_b, (f_i, g_j)], \varepsilon, l)$
6. (3.) \wedge (5.) \wedge Definícia δ' \rightsquigarrow Existuje výpočet $T_w^{M'}$ stroja M' na slove w , ktorý je znázornený na obrázku 3.4. Listy $T_w^{M'}$ sú označené akceptačnou konfiguráciou a ich synchronizačná postupnosť je rovnaká.
7. (6.) $\rightsquigarrow w \in L(M')$

$$\begin{array}{ccc}
\vdots & & \vdots \\
([q, (f_i, g_j)], \varepsilon, k) & & ([q'_b, (f_i, g_j)], \varepsilon, l) \\
\downarrow 3.3 & & \downarrow 3.6 \\
([q'_f, 2], \varepsilon, k) & & ([q'_f, 2], \varepsilon, l) \\
\downarrow 3.7 & & \downarrow 3.7 \\
(q'_f, \varepsilon, k) & & (q_f, \varepsilon, l)
\end{array}$$

Obrázok 3.4: SAPA simulujúci P2A, akceptačný výpočet

$$L(M') \subset L(M)$$

Nech $w \in L(M')$ je ľubovoľné. Čiže existuje výpočet stroja $L(M')$ na slove w . Podľa lemy 3.2 je výpočet tvaru znázorneného na obrázku 3.3 a existuje výpočet stroja M na slove w , ktorý obsahuje konfiguráciu $(p_m, \varepsilon, k_m, l_m)$, kde $p_m \in F$. Z toho vyplýva, že $w \in L(M)$, čo sme chceli dokázať. \square

Poznámka 3.1. *Technika použitá v predchádzajúcej konštrukcii sa dá rozšíriť na simulovanie automatu s k počítadlami, kde k je ľubovoľná konštanta. Proces sa na začiatku rozdelí na k procesov. Každý proces bude simulovať jedno počítadlo. Synchronizačné symboly budú obsahovať k zložiek. Každá zložka bude obsahovať informáciu o stave počítadla pred krokom výpočtu a informáciu o aktualizácii počítadla po kroku výpočtu.*

Nasledovná veta hovorí, že synchronizovane alternujúce automaty s počítadlom charakterizujú triedu rekurzívne vyčísliteľných jazykov. To inými slovami znamená, že všetko čo sa dá vypočítať, sa dá vypočítať aj na synchronizovane alternujúcich automatoch s počítadlom.

Veta 3.2. $\mathcal{L}_{RE} = \mathcal{L}_{SAPA}$

Dôkaz. $\mathcal{L}_{RE} \subset \mathcal{L}_{SAPA}$

Pre každý jazyk z \mathcal{L}_{RE} existuje Turingov stroj A , ktorý ho akceptuje. Vieme, že pre A existuje ekvivalentný automat s dvomi počítadlami B . Podľa vety 3.1 existuje ekvivalentný SAPA C s B . Čiže C je ekvivalentný aj s A . Čo sme chceli dokázať.

$$\mathcal{L}_{SAPA} \subset \mathcal{L}_{RE}$$

Vyšplýva z Turingovej tézy. \square

4 Počet synchronizačných symbolov

V tejto kapitole sa budeme zaoberať synchronizovane alternujúcimi modelmi vo všeobecnosti, bez udania sekvenčného modelu, z ktorého vychádza. Preskúmame, ako počet synchronizačných symbolov ovplyvňuje výpočtovú silu synchronizovane alternujúceho zariadenia.

4.1 Alternovanie s jedným synchronizačným symbolom

V nasledovnom tvrdení ukážeme, že synchronizovanie len v jednom synchronizačnom symbole alternujúcemu zariadeniu vôbec nepomôže. Intuitívne, keď proces synchronizuje, tak nevie či aj iný proces synchronizuje, alebo už prešiel do akceptačného stavu. Vie len, že keby bol iný proces ešte živý a chcel synchronizovať, tak bude synchronizovať v rovnakom symbole. Keďže je symbol len jeden, tak táto informácia je skutočne triviálna.

Lema 4.1. *Nech A je SAZ, ktoré synchronizuje len v jednom synchronizačnom symbole. Potom existuje alternujúce zariadenie B , ktorého model vychádza z rovnakého sekvenčného automatového modelu a akceptuje rovnaký jazyk.*

Dôkaz. Nech B je definované rovnako ako A , len synchronizačné stavy nebudú synchronizačné. T.j. ak A je $(Q, S, A, \Sigma, \Gamma, \delta, q_0, F)$, tak B je $(Q \cup (Q \times \{1\}), A, \Sigma, \Gamma, \delta, q_0, F)$

Pre ľubovoľné slovo u platí, že úplný strom konfigurácii T_u^A je rovnaký ako T_u^B , pretože A a B majú rovnaké prechodové funkcie.

$$L(A) \subset L(B)$$

Nech $w \in L(A)$, čiže existuje T_w^A podstrom T_w^A a platí:

1. Každý uzol z T_w^A označený univerzálnou konfiguráciou má rovnakých priamych nasledovníkov ako na T_w^A , alebo nemá žiadneho nasledovníka
2. Každý uzol z T_w^A označený existenčnou konfiguráciou má najviac jedného priameho nasledovníka
3. Pre každý uzol t_1 a t_2 platí, že $SP(t_1)$ je počiatočnou podpostupnosťou $SP(t_2)$ alebo naopak
4. T_w^A je konečný
5. Každý list je označený akceptačnou konfiguráciou

Platí $T_w^A = T_w^B$.

Nech $T_w'^B = T_w'^A$. Potom platí:

1. Každý uzol z $T_w'^B$ označený univerzálnou konfiguráciou má rovnakých priamych nasledovníkov ako na T_w^B , alebo nemá žiadneho nasledovníka
2. Každý uzol z $T_w'^B$ označený existenčnou konfiguráciou má najviac jedného priameho nasledovníka
3. $T_w'^B$ je konečný
4. Každý list je označený akceptačnou konfiguráciou

Čiže podľa definície alternujúceho stroja slovo w patrí do $L(B)$.

$$L(B) \subset L(A)$$

Nech $w \in L(B)$, čiže existuje $T_w'^B$ podstrom T_w^B a platí:

1. Každý uzol z $T_w'^B$ označený univerzálnou konfiguráciou má rovnakých priamych nasledovníkov ako na T_w^B , alebo nemá žiadneho nasledovníka
2. Každý uzol z $T_w'^B$ označený existenčnou konfiguráciou má najviac jedného priameho nasledovníka
3. $T_w'^B$ je konečný
4. Každý list je označený akceptačnou konfiguráciou

Platí $T_w^B = T_w^A$.

Nech $T_w'^A = T_w'^B$. Potom platí:

1. Každý uzol z $T_w'^A$ označený univerzálnou konfiguráciou má rovnakých priamych nasledovníkov ako na T_w^A , alebo nemá žiadneho nasledovníka
2. Každý uzol z $T_w'^A$ označený existenčnou konfiguráciou má najviac jedného priameho nasledovníka
3. $T_w'^A$ je konečný
4. Každý list je označený akceptačnou konfiguráciou

Každý uzol má synchronizačnú postupnosť prázdnu, alebo obsahuje len symboly 1. Čiže triviálne platí, že pre každý uzol t_1 a t_2 $SP(t_1)$ je počiatočnou podpostupnosťou $SP(t_2)$ alebo naopak. Čiže podľa definície synchronizovane alternujúceho stroja slovo w patrí do $L(A)$. \square

Lema 4.2. *Nech B je alternujúce zariadenie. Potom existuje SAZ A , ktorého model vychádza z rovnakého sekvenčného automatového modelu a akceptuje rovnaký jazyk.*

Dôkaz. SAZ je rozšírenie modelu alternujúceho zariadenia. A môže byť SAZ, ktoré vôbec nesynchronizuje a je rovnaké ako B . Triviálne $L(A) = L(B)$. \square

Veta 4.1. *Trieda jazykov definovaná synchronizovane alternujúcimi zariadeniami, ktoré používajú len jeden synchronizačný symbol je rovnaká, ako trieda jazykov definovaná alternujúcimi zariadeniami, ktorých model vychádza z rovnakého sekvenčného automatového modelu.*

Dôkaz. Vyplýva z predchádzajúcich pomocných tvrdení. \square

4.2 Alternovanie s dvomi synchronizačnými symbolmi

V tejto kapitole sa budeme zaoberať otázkou, akú výpočtovú silu majú zariadenia, ktoré používajú len dva synchronizačné symboly. Ukážeme že rovnakú ako zariadenia, ktoré používajú neohraničený počet synchronizačných symbolov. Jadrom simulácie zariadenia, ktoré používa ľubovoľný konečný počet synchronizačných symbolov, je zakódovanie symbolov do binárnej sústavy.

Lema 4.3. *Nech A je ľubovoľné SAZ, ktoré pri výpočte používa k synchronizačných symbolov. Potom existuje SAZ B , ktorého sekvenčný automatový model je rovnaký a ktoré pri výpočte používa 2 synchronizačné symboly. Navyše ak A je $T(n)$ časovo ohraničený, tak B je $(k+1) \cdot T(n)$ časovo ohraničený.*

Dôkaz. Ak $k \leq 2$ tak tvrdenie triviálne platí. Zaoberajme sa preto situáciou, keď $k > 2$.

Nech A je definovaná ako $(Q_A, S_A, A_A, \Sigma, \Gamma, \delta_A, q_0, F)$.

Synchronizačné symboly zoradíme do postupnosti. Keďže synchronizačných symbolov je konečne veľa, postupnosť je konečná. Nech $f : S_A \rightarrow \mathbb{N}$ je taká funkcia, že $f(x)$ je poradie symbolu x v tejto postupnosti.

B bude rovnaké SAZ len ak A chcelo synchronizovať v symbole x , tak B bude synchronizovať postupne v symboloch $1^{f(x)}2$. Formálne:

Keďže chceme dokazovať tvrdenie pre všetky možné SAZ, musíme sa vysporiadať s ich možnými rozdielmi. Je rozumné uvažovať len zariadenia, ktoré vidia zo vstupného slova jeden znak. Zariadenia, ktoré vidia viac znakov, môžeme nahradiť zariadeniami, ktoré majú v stave buffer. O pamäti budeme uvažovať abstraktne. Vieme to, že v konfigurácii procesu sa môže vyskytovať neohraničená pamäť, ale pri relácii krok výpočtu sa berie v úvahu len jej konečná časť. Konečná časť pamäte sa modifikuje na konečnú časť pamäte, ostatné časti pamäte ostanú nezmenené. Takýchto konečných častí pamäte musí byť konečne veľa. Označme konečnú množinu takýchto častí pamätí $\mathbb{M} \subset_{kon} \Gamma^*$. Dodajme len, že ak zariadenie nemá pamäť (konečný automat), tak $\mathbb{M} = \emptyset$. Ak to zhrnieme, tak:

$$\delta_A : Q'_A \times \Sigma_\epsilon \times \mathbb{M} \rightarrow 2^{Q'_A \times \mathbb{M}}$$

Nech $|S_A| = k$. Pre jednoduchosť nech $S_A = \mathbb{N}_k$.

Nech B je $(Q_B, S_B, A_B, \Sigma, \Gamma, \delta_B, q_0, F)$

$$\begin{aligned} Q_B &= Q_A \cup (Q_A \times \mathbb{N}_k \times \mathbb{N}_k) \\ S_B &= \{1, 2\} \\ A_B &= \{q \mid q \in Q_A \wedge q \in A_A\} \\ &\quad \cup \{[(q, l, 1), 2] \mid q \in Q_A \wedge l \in S_A \wedge [q, l] \in A_A\} \end{aligned}$$

δ_B bude definovaná nasledovne:

$$(\forall p, q \in Q_A) (\forall a \in \Sigma_\varepsilon) (\forall g, h \in M) (\forall r, t \in \mathbb{N}_k) (\forall s \in \mathbb{N}_{k-1}) (s \leq r) :$$

$$\begin{aligned} (p, g) \in \delta_A(q, a, h) &\rightarrow (p, g) \in \delta_B(q, a, h) \\ ([p, r], g) \in \delta_A(q, a, h) &\rightarrow ([p, r, 1], g) \in \delta_B(q, a, h) \\ \delta_B([p, r, s], 1, \varepsilon, g) &= \{([p, r, s+1], 1, g)\} \\ \delta_B([p, r, r], 1, \varepsilon, g) &= \{([p, r, r], 2, g)\} \\ (p, g) \in \delta_A([q, r], a, h) &\rightarrow (p, g) \in \delta_B([q, r, r], 2, a, h) \\ ([p, t], g) \in \delta_A([q, r], a, h) &\rightarrow ([p, t, 1], g) \in \delta_B([q, r, r], 2, a, h) \end{aligned}$$

Je zrejmé, že zariadenia A a B akceptujú rovnaký jazyk. □

Veta 4.2. *Trieda jazykov definovaná synchronizovane alternujúcimi zariadeniami, ktoré používajú dva synchronizačné symboly je rovnaká, ako trieda jazykov definovaná synchronizovane alternujúcimi zariadeniami, ktorých model vychádza z rovnakého sekvenčného automatového modelu.*

Dôkaz. Je zrejmé, že trieda definovaná synchronizovane alternujúcimi zariadeniami je väčšia alebo rovnaká, ako trieda definovaná synchronizovane alternujúcimi zariadeniami, ktoré vychádzajú z rovnakého sekvenčného modelu a používajú len 2 synchronizačné symboly.

Druhá inklúzia vyplýva z predchádzajúceho tvrdenia. □

5 Alternujúce konečné automaty s jedným synchronizačným symbolom a zarážkou

V tejto kapitole sa budeme zaoberať myšlienkou použitia konečných automátov, ktoré používajú len jeden synchronizačný symbol, ale v akceptačnom stave môžu komunikovať s druhým synchronizačným symbolom. Po použití tohto druhého symbolu proces už vo výpočte nemôže pokračovať. Ak dočítal slovo, tak je tento proces v akceptačnej konfigurácii. Inak je v konfigurácii, ktorá nie je akceptačná.

Formálne je tento model definovaný v definícii 3 v kapitole 2.

5.1 Porovnanie s Chomského hierarchiou

Najskôr ukážeme, že toto zariadenie je výpočtovo silnejšie ako jednoduchý alternujúci konečný automat. Podľa [6] alternujúci konečný automat je rovnako výpočtovo silný ako klasický konečný automat a teda trieda jazykov nimi definovaná je trieda regulárnych jazykov.

Teraz ukážeme, že existuje S_2^1 AKA, ktorý rozpoznáva jazyk, na ktorý nám nestačia ani bezkontextové prostriedky. Konkrétne jazyk $a^n b^n c^n$.

Príklad 5.1. *Nech S_2^1 AKA M je $(Q, A, \Sigma, \delta, q_0, F)$, kde:*

$$\begin{aligned} Q &= \{q_0, q_1, q_2, q_3, q_a, q_b, q_c, q_f\} \\ A &= \{q_0\} \\ \Sigma &= \{a, b, c\} \\ F &= \{q_f, [q_f, 2]\} \end{aligned}$$

Funkcia δ bude definovaná nasledovne:

$$\delta(q_0, a) = \{[q_a, 1], q_b, q_c, q_1\} \quad (5.1)$$

$$\delta([q_a, 1], a) = \{[q_a, 1]\} \quad (5.2)$$

$$\delta([q_a, 1], b) = \{q_a\} \quad (5.3)$$

$$\delta(q_a, b) = \{q_a\} \quad (5.4)$$

$$\delta(q_a, c) = \{q_a\} \quad (5.5)$$

$$\delta(q_a, \varepsilon) = \{[q_f, 2]\} \quad (5.6)$$

$$\delta(q_b, a) = \{q_b\} \quad (5.7)$$

$$\delta(q_b, c) = \{q_b\} \quad (5.8)$$

$$\delta(q_b, b) = \{[q_b, 1]\} \quad (5.9)$$

$$\delta(q_b, \varepsilon) = \{[q_f, 2]\} \quad (5.10)$$

$$\delta([q_b, 1], b) = \{[q_b, 1]\} \quad (5.11)$$

$$\delta([q_b, 1], c) = \{q_b\} \quad (5.12)$$

$$\delta(q_c, a) = \{q_c\} \quad (5.13)$$

$$\delta(q_c, b) = \{q_c\} \quad (5.14)$$

$$\delta(q_c, c) = \{[q_c, 1]\} \quad (5.15)$$

$$\delta([q_c, 1], c) = \{[q_c, 1]\} \quad (5.16)$$

$$\delta([q_c, 1], \varepsilon) = \{[q_f, 2]\} \quad (5.17)$$

$$\delta(q_1, a) = \{q_1\} \quad (5.18)$$

$$\delta(q_1, b) = \{q_2\} \quad (5.19)$$

$$\delta(q_2, b) = \{q_2\} \quad (5.20)$$

$$\delta(q_2, c) = \{q_3\} \quad (5.21)$$

$$\delta(q_3, c) = \{q_3\} \quad (5.22)$$

$$\delta(q_3, \varepsilon) = \{q_f\} \quad (5.23)$$

Z definície automatu M vyplýva, že každý jeho výpočet začína rozdelením koreňa na vlákna, ktorých prvé vrcholy začínajú stavmi $[q_a, 1]$, q_b , q_c , q_1 . Označme tieto vlákna symbolmi A , B , C , D . Iné vlákna už vo výpočte nemôžu vzniknúť.

Pred samotným dôkazom korektnosti zariadenia dokážeme dve pomocné tvrdenia. Prvé pomocné tvrdenie hovorí o invariante platnom v prvých troch procesoch. Druhé hovorí o vlastnosti, ktorú overí štvrtý proces.

Lema 5.1. *Nech T_w^M je ľubovoľný výpočet zariadenia M na ľubovoľnom slove w . Potom pre každý vrchol v vlákna A, B, C platí, že počet symbolov 1 v $SP(v)$ sa rovná počtu prečítaných symbolov a, b, c zo vstupného slova na ceste od koreňa po vrchol v .*

Dôkaz. Dôkaz urobíme matematickou indukciou vzhľadom na m - vzdialenosť vrcholu od koreňa výpočtu T_w^M .

1° $m = 1$

Prvý krok výpočtu je deterministický, takže nám stačí rozobrať tri prípady, z ktorých dva sú symetrické.

Pre proces A : Zo vstupu je prečítaný jeden symbol a a synchronizačná postupnosť obsahuje jeden symbol 1.

Pre procesy B, C : Zo vstupu nie je prečítaný žiadny symbol b, c a synchronizačná postupnosť je prázdna.

2° Nech tvrdenie platí pre $m \succeq 1$. Ukážme, že platí aj pre $m + 1$.

Nech u je vrchol, ktorého vzdialenosť od koreňa je $m + 1$. Nech v je jeho priamy nasledovník.

Z definície δ funkcie je zrejmé, že vrchol vlákna A, B, C môže byť označený len konfiguráciou so stavom z množiny $\{[q_a, 1], q_a, [q_f, 2]\}, \{q_b, [q_b, 1], [q_f, 2]\}, \{q_c, [q_c, 1], [q_f, 2]\}$. Navyše vrcholy so stavom z množiny $\{[q_a, 1], q_a\}, \{q_b, [q_b, 1]\}, \{q_c, [q_c, 1]\}$ sa vyskytujú len vo vlákne A, B, C .

Prípady, že $c(v)$ má stav $[q_f, 2]$ nemusíme rozoberať, pretože podľa definície 3 tento vrchol už nemôže mať nasledovníkov. Čiže nám stačí rozobrať šesť prípadov na stav konfigurácie $c(v)$.

Nech $p_x(y)$ je počet prečítaných symbolov x na ceste od koreňa po vrchol y . Potom indukčné tvrdenie môžeme prepísať nasledovne.

$$\begin{aligned} v \in A & \rightsquigarrow \#_1(SP(v)) = p_a(v) \rightsquigarrow \#_1(SP(u)) = p_a(u) \\ v \in B & \rightsquigarrow \#_1(SP(v)) = p_b(v) \rightsquigarrow \#_1(SP(u)) = p_b(u) \\ v \in C & \rightsquigarrow \#_1(SP(v)) = p_c(v) \rightsquigarrow \#_1(SP(u)) = p_c(u) \end{aligned} \quad (5.24)$$

Rozoberme teraz všetky možné prípady.

1. $u, v \in A$, $c(v)$ má stav $[q_a, 1]$

- $v \rightarrow_{5.2} u$, t.j. $([q_a, 1], ar) \vdash ([q_a, 1], r)$
 $p_a(u) = p_a(v) + 1, \#_1(SP(u)) = \#_1(SP(v)) + 1$
- $v \rightarrow_{5.3} u$, t.j. $([q_a, 1], br) \vdash (q_a, r)$
 $p_a(u) = p_a(v), \#_1(SP(u)) = \#_1(SP(v))$

2. $u, v \in A$, $c(v)$ má stav q_a

- $v \rightarrow_{5.4} u$, t.j. $(q_a, br) \vdash (q_a, r)$
 $p_a(u) = p_a(v)$, $\#_1(SP(u)) = \#_1(SP(v))$
- $v \rightarrow_{5.5} u$, t.j. $(q_a, cr) \vdash (q_a, r)$
 $p_a(u) = p_a(v)$, $\#_1(SP(u)) = \#_1(SP(v))$
- $v \rightarrow_{5.6} u$, t.j. $(q_a, r) \vdash ([q_f, 2], r)$
 $p_a(u) = p_a(v)$, $\#_1(SP(u)) = \#_1(SP(v))$

3. $u, v \in B$, $c(v)$ má stav q_b

- $u, v \in A$, $c(v)$ má stav q_b
- $v \rightarrow_{5.7} u$, t.j. $(q_b, ar) \vdash (q_b, r)$
 $p_b(u) = p_b(v)$, $\#_1(SP(u)) = \#_1(SP(v))$
- $v \rightarrow_{5.8} u$, t.j. $(q_b, cr) \vdash (q_b, r)$
 $p_b(u) = p_b(v)$, $\#_1(SP(u)) = \#_1(SP(v))$
- $v \rightarrow_{5.9} u$, t.j. $(q_b, br) \vdash ([q_b, 1], r)$
 $p_b(u) = p_b(v) + 1$, $\#_1(SP(u)) = \#_1(SP(v)) + 1$
- $v \rightarrow_{5.10} u$, t.j. $(q_b, r) \vdash ([q_f, 2], r)$
 $p_b(u) = p_b(v)$, $\#_1(SP(u)) = \#_1(SP(v))$

4. $u, v \in B$, $c(v)$ má stav $[q_b, 1]$

- $v \rightarrow_{5.11} u$, t.j. $([q_b, 1], br) \vdash ([q_b, 1], r)$
 $p_b(u) = p_b(v) + 1$, $\#_1(SP(u)) = \#_1(SP(v)) + 1$
- $v \rightarrow_{5.12} u$, t.j. $([q_b, 1], cr) \vdash (q_b, r)$
 $p_b(u) = p_b(v)$, $\#_1(SP(u)) = \#_1(SP(v))$

5. $u, v \in C$, $c(v)$ má stav q_c

- $u, v \in A$, $c(v)$ má stav q_c
- $v \rightarrow_{5.13} u$, t.j. $(q_c, ar) \vdash (q_c, r)$
 $p_c(u) = p_c(v)$, $\#_1(SP(u)) = \#_1(SP(v))$
- $v \rightarrow_{5.14} u$, t.j. $(q_c, br) \vdash (q_c, r)$
 $p_c(u) = p_c(v)$, $\#_1(SP(u)) = \#_1(SP(v))$
- $v \rightarrow_{5.15} u$, t.j. $(q_c, cr) \vdash ([q_c, 1], r)$
 $p_c(u) = p_c(v) + 1$, $\#_1(SP(u)) = \#_1(SP(v)) + 1$

6. $u, v \in C$, $c(v)$ má stav $[q_c, 1]$

- $v \xrightarrow{5.16} u$, t.j. $([q_c, 1], cr) \vdash ([q_c, 1], r)$
 $p_c(u) = p_c(v) + 1$, $\#_1(SP(u)) = \#_1(SP(v)) + 1$
- $v \xrightarrow{5.17} u$, t.j. $([q_c, 1], r) \vdash ([q_f, 2], r)$
 $p_c(u) = p_c(v)$, $\#_1(SP(u)) = \#_1(SP(v))$

Pre všetky možné prípady implikácie 5.24 platia a teda platí indukčný krok.

3° Z prvého a druhého kroku indukcie vyplýva, že tvrdenie platí pre každé m , pre ľubovoľný vrchol výpočtu T_w^M . \square

Lema 5.2. *Nech T_w^M je ľubovoľný výpočet zariadenia M na ľubovoľnom slove w . Potom vlákno D končí listom označeným akceptačnou konfiguráciou len ak $w \in \{a^i b^j c^k \mid i, j, k \geq 1\}$. Navyše synchronizačná postupnosť listu vlákna D je prázdna.*

Dôkaz. Pozrime sa ako vyzerá výpočet vo vlákne D .

1. Zo stavu q_0 na vstupný symbol a do stavu q_1 vznikne vlákno $D \wedge$
 \wedge Stav q_0 nie je akceptačný $\rightsquigarrow w \in a\Sigma^*$
2. Z q_1 existujú prechody len na znaky a – do stavu q_1 , b – do stavu $q_2 \wedge$
 \wedge Stav q_1 nie je akceptačný $\rightsquigarrow w \in a^+b\Sigma^*$
3. Z q_2 existujú prechody len na znaky b – do stavu q_2 , c – do stavu $q_3 \wedge$
 \wedge Stav q_2 nie je akceptačný $\rightsquigarrow w \in a^+b^+c\Sigma^*$
4. Z q_3 existujú prechody len na znak c – do stavu q_3 , na ε – do stavu q_f
 \wedge Stav q_3 nie je akceptačný $\rightsquigarrow w \in a^+b^+c^+\Sigma^*$
5. Zo stavu q_f už prechody neexistujú \wedge Aby výpočet mohol byť akceptačný, vstupné slovo musí byť už celé prečítané \wedge
 \wedge Stav q_f je akceptačný $\rightsquigarrow w \in a^+b^+c^+$

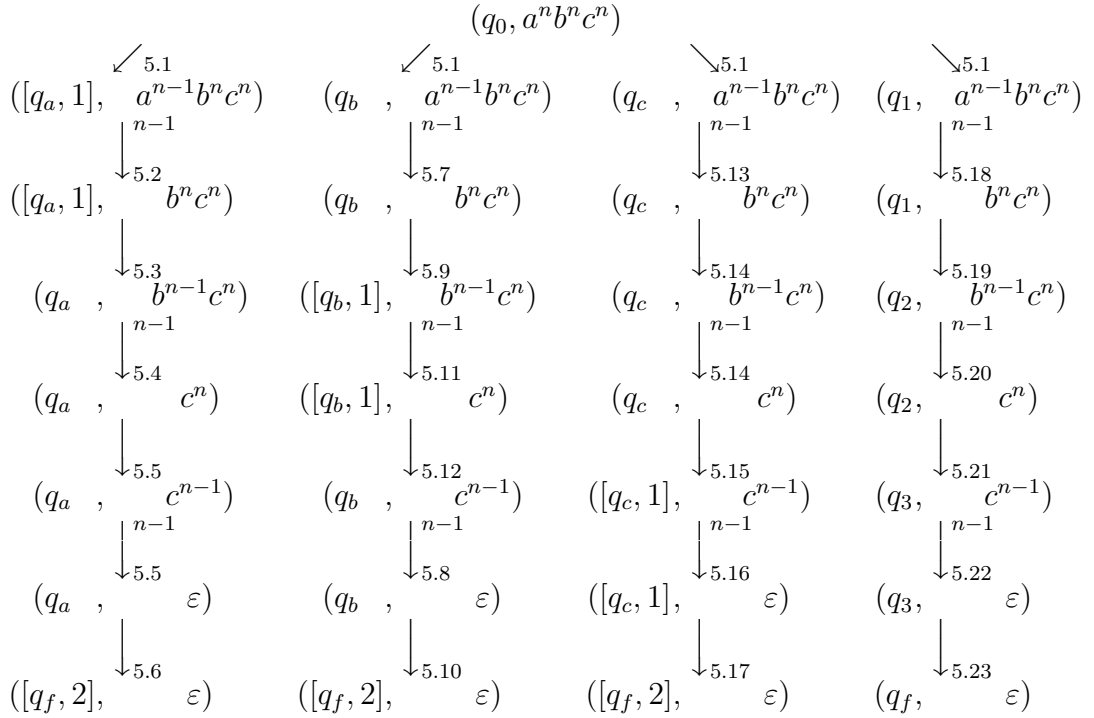
\square

Lema 5.3. $\{a^n b^n c^n | n \geq 1\} \in \mathcal{L}_{S_2^1 AKA}$

Dôkaz. $\{a^n b^n c^n | n \geq 1\} \subset L(M)$

Zoberme ľubovoľné $n \geq 1$, nech $w = a^n b^n c^n$. Nech T_w^M je výpočet automatu M na slove w , ktorý je znázornený na obrázku 5.1. $v \rightarrow_l^k u$ je skratka za cestu z vrcholu označeného konfiguráciou v do vrcholu označeného konfiguráciou u , ktorá vznikla k -násobným použitím pravidla l . Pre $k = 1$ je horný index vynechaný. Na obrázku je značka zakreslená vo zvislom smere.

Keďže všetky listy sú označené akceptačnou konfiguráciou, z lemy 5.1 vyplýva, že synchronizačná postupnosť prvých troch listov je rovnaká, z lemy 5.2 vyplýva, že synchronizačná postupnosť štvrtého listu je prázdna, výpočet T_w^M je akceptačný.



Obrázok 5.1: Výpočet na slove $a^n b^n c^n$

$$L(M) \subset \{a^n b^n c^n \mid n \geq 1\}$$

1. Nech $w \in L(M)$. To znamená, že existuje akceptačný výpočet T_w^M .
2. (1.) \wedge Lema 5.2 $\rightsquigarrow w \in \{a^i b^j c^k \mid i, j, k \geq 1\}$
3. Nech v_a, v_b, v_c sú listy vlákien A, B, C . Keďže sú označené akceptačnou konfiguráciou, prečítali celé vstupné slovo.
4. (3.) \wedge Lema 5.1

$$\#_1(SP(v_a)) = \#_a(w)$$

$$\#_1(SP(v_b)) = \#_b(w)$$

$$\#_1(SP(v_c)) = \#_c(w)$$

5. (4.) \wedge M je S_2^1 AKA

$$SP(v_a) = 1^{\#_a(w)} 2$$

$$SP(v_b) = 1^{\#_b(w)} 2$$

$$SP(v_c) = 1^{\#_c(w)} 2$$

6. Ľahko vidieť

$$1^i 2 \prec 1^j 2 \vee 1^j 2 \prec 1^i 2 \rightsquigarrow i = j$$

7. (5.) \wedge (6.) $\rightsquigarrow w \in \{w \mid \#_a(w) = \#_b(w) = \#_c(w)\}$
8. (2.) \wedge (7.) $\rightsquigarrow w \in \{a^n b^n c^n \mid n \geq 1\}$

□

Nasledovné tvrdenie nám dovoľuje čiastočne zaradiť nami definovanú triedu do Chomského hierarchie. Okrem iného hovorí, že takto definované synchronizované alternovanie pridáva konečným automatom výpočtovú silu.

Veta 5.1. $\mathcal{L}_{S_2^1AKA} \cap \mathcal{L}_{ECS} = \mathcal{L}_{CF} \neq \emptyset$

Dôkaz. Vyplýva z lemy 5.3. □

Keďže skúmaný model určite nie je silnejší ako model synchronizovane alternujúcich konečných automatov vieme, že je určite pod triedou kontextových jazykov. My ale vyslovíme hypotézu, že dokonca tak jednoduché zariadenie ako je automat s počítadlom, dokáže akceptovať jazyk, pre ktorý neexistuje alternujúci konečný automat synchronizujúci s jedným symbolom a záťažkou. Ako kandidát pre takýto jazyk by mohol byť jazyk dobre uzátvorkovaných výrazov pre jednu dvojicu zátvoriek – D_1 .

Hypotéza 5.1. $\mathcal{L}_{PA} = \mathcal{L}_{S_2^1AKA} \neq \emptyset$

5.2 Rovnako synchronizujúci normálny tvar

Lema 5.4. *Pre každý S_2^1 AKA M a každé prirodzené číslo k existuje S_2^1 AKA M' taký, že $L(M) = L(M')$ a každý list každého akceptačného výpočtu zariadenia M' je označený rovnakým stavom a má synchronizačnú postupnosť, ktorá obsahuje aspoň k symbolov 1 a jeden symbol 2, t.j. všetky synchronizujúce postupnosti listov sú $1^{k+l}2$ pre vhodné $l \geq 0$.*

Dôkaz. Nech M je $(Q, A, \Sigma, \delta, q_0, F)$.

Zariadenie, ktoré bude simulovať M bude pracovať nasledovne: Najskôr bude k -krát synchronizovať. Následne začne simulovať všetky procesy automatu M , ktoré synchronizovali menej ako k -krát v jednom koreňovom procese. Simulácia bude prebiehať po krokoch, ktoré predstavujú prečítanie jedného písmena zo vstupu všetkými procesmi. Ak nejaký proces P bude chcieť synchronizovať viacej ako k -krát, vytvorí sa nový proces P' , ktorý bude v simulácii procesu P pokračovať. P' bude rovnaký ako P s výnimkou, ak P chcel prejsť do akceptačného stavu. Vtedy P' bude ešte niekoľkokrát synchronizovať v symboloch 1 a následne prejde do akceptačného stavu $[q_f, 2]$. To zabezpečí, že všetky procesy budú synchronizovať v rovnakej synchronizačnej postupnosti. Ak koreňový proces prečíta celé vstupné slovo, tak overí či môže akceptovať. To urobí nasledovne: Ak vytvoril nový proces, tak žiadny ním simulovaný proces nemôže synchronizovať v symbole 2. Ak nevytvoril nový proces, tak niektoré zo simulovaných procesov môžu synchronizovať v symbole 2, no musia predtým synchronizovať v rovnakom počte symbolov 1 a žiadny z procesov, ktoré nesynchronizujú v symbole 2, nemôže synchronizovať v symboloch 1 viac krát.

Formálne :

Nech δ_ε je funkcia definovaná nasledovne: $\delta_\varepsilon(q) = \{(p_1, s_1), (p_2, s_2), \dots, (p_n, s_n)\}$, ak z vrcholu v označeného konfiguráciou (q, ε) možno na M urobiť výpočet, ktorý obsahuje listy v_1, v_2, \dots, v_n . Vrchol v_i je označený konfiguráciou so stavom $p_i \in Q_1$ ($Q_1 = Q \cup (Q \times \{1\})$) a postupnosť synchronizačných symbolov na ceste z vrcholu v do v_i je sp_i a $s_i = \#_1(sp_i)$, $s_i \leq k$.

Nech M' je $(Q', A', \Sigma, \delta', q'_0, F')$, kde:

$$\begin{aligned}
Q' &= 2^{(Q \times \mathbb{N}_{0,k}) \cup \{q'_{ok}, b\}} \cup \{q'_i \mid i \in \mathbb{N}_{0,k-1}\} \cup Q \cup \{q'_f\} \\
A' &= \{B \in Q' \mid \exists q \in Q_1, (q, k) \in B\} \\
F' &= \{q'_f\} \\
\delta'(q'_0, \varepsilon) &= \{[q'_1, 1]\} \\
\delta'([q'_i, 1], \varepsilon) &= \{[q'_{i+1}, 1]\}, i \in \mathbb{N}_{k-2} \\
\delta'([q'_k, 1], \varepsilon) &= \{\{(q_0, 0)\}\} \\
\delta'(B, a) &= Krok(B, a) \cup PoslednyKrok(B, a), \\
&\quad B \in 2^{(Q \times \mathbb{N}_{0,k-1}) \cup \{b\}}, a \in \Sigma \\
\delta'(B, \varepsilon) &= \{\{(q, i) \mid (q, i) \in B, i \neq k\} \cup \{b\}\} \cup \\
&\quad \cup \{q \mid (q, k) \in B\}, B \in 2^{(Q \times \mathbb{N}_{0,k}) \cup \{b\}}, \exists q \in Q_1((q, k) \in B) \\
\delta'(q, a) &= \delta(q, a) \\
&\quad \cup \{[q'_f, 2] \mid (F \cup (F \times \{2\})) \cap \delta(q, a) \neq \emptyset\} \cup \\
&\quad \cup \{\{[q'_{ok}, b], 1\} \mid F \cap \delta(q, a) \neq \emptyset\} - \\
&\quad - (Q \times \{2\}), \text{ ak } q \in Q_1 \\
\delta'(\{[q'_{ok}, b], 1\}, \varepsilon) &= \{\{[q'_{ok}, b], 1\}, [q'_f, 2]\}
\end{aligned}$$

Koreňový proces počas výpočtu potrebuje vedieť, či už vytvoril nový proces, alebo ešte nie. Túto informáciu uchovávame v prítomnosti prvku b v množine tvoriacej stav koreňového procesu.

V definícii funkcie δ' sú použité množiny *Krok* a *PoslednyKrok*. Keďže definícia týchto množín je zložitá a pri zápise v štandardnej notácii by bol jej zápis ťažko čitateľný, budú definované ako množiny výsledkov nedeterministických funkcií *krok* a *poslednyKrok*. Zvolili sme zápis podobný notácii programovacieho jazyka Pascal:

```

function krok( $B, a$ )
begin
   $Result := \emptyset$ ;
   $error := 0$ ;
  foreach(( $p, e$ )  $\in B$ )
  begin
    foreach(( $r, f$ )  $\in \delta_\varepsilon(p)$ )
    begin
      if  $e + f > k$  then  $error := 1$ ;
      else if  $e + f = k$  then begin
        if  $r \in Q \times \{1\}$  then  $Result := Result \cup \{(r, k)\}$  else  $error := 1$ ;
      end else begin
        if  $r \in A$  then  $S := \delta(r, a)$  else  $S := random(\delta(r, a))$ ;
        if  $S = \emptyset$  then  $error := 1$ ;
        foreach( $s \in S$ )
        begin
          if  $s \in Q \times \{1\}$  then  $g := 1$  else  $g := 0$ ;
           $Result := Result \cup \{(s, e + f + g)\}$ ;
        end
      end
    end
  end
  if  $b \in B$  then  $Result := Result \cup \{b\}$ ;
  if  $error = 1$  then  $Result := \emptyset$ ;
end

```

Význam premenných vo funkcii *krok*:

- $Result$ - Postupnosť dvojíc (stav, počet syn. symbolov)
simulovaných procesov po prečítaní znaku a
- $error$ - Pri simulácii kroku nastala chyba, výpočet sa musí zaseknúť
- p, r, s - Aktuálny stav
- e, f, g - Čiastkový počet symbolov 1 v aktuálnej synchronizačnej
postupnosti
- S - Zoznam aktuálnych stavov po prechode na znak a

Funkcia *krok* používa funkciu *random*, ktorá vráti jednoprvkovú množinu, ktorá obsahuje nedeterministicky vybraný prvok zo vstupnej množiny.

```

function poslednyKrok( $B, a$ ) begin
  error := 0;
  syn := -1;
  pSyn := -1;
  foreach(( $p, e$ )  $\in B$ ) begin
    foreach(( $r, f$ )  $\in \delta_\varepsilon(p)$ ) begin
      if  $e + f > k$  then error := 1;
      else if  $e + f = k$  then begin
        if  $r \in Q \times \{1\}$  then begin
          Result := Result  $\cup \{(r, k)\}$ ;
           $B := B \cup \{b\}$ ;
        end else error := 1;
      end else begin
        if  $r \in A$  then  $S := \delta(r, a)$  else  $S := \text{random}(\delta(r, a))$ ;
        if  $S = \emptyset$  then error := 1;
        foreach( $s \in S$ ) begin
          if  $s \in Q \times \{2\}$  then begin
            if  $s \notin F$  then error := 1;
            if syn = -1 then begin
              if  $pSyn > e + f$  then error := 1;
              syn :=  $e + f$ ;
            end else if syn  $<> e + f$  then error := 1;
          end else begin
            if  $s \in Q \times \{1\}$  then  $g := 1$  else  $g := 0$ ;
            if  $s \in F$  and  $\text{random}(\{0, 1\}) = \{0\}$  then begin
              if syn = -1 then  $pSyn := \text{Max}(pSyn, e + f + g)$ ;
              else if syn  $< e + f + g$  then error := 1;
            end else foreach(( $t, h$ )  $\in \delta_\varepsilon(s)$ ) begin
              if  $e + f + g + h > k$  then error := 1;
              else if  $e + f + g + h = k$  then begin
                if  $t \in Q \times \{1\}$  then begin
                  Result := Result  $\cup \{(t, k)\}$ ;
                   $B := B \cup \{b\}$ ;
                end else error := 1;
              end else begin

```

```

end else begin
  if  $t \in Q \times \{2\}$  then begin
    if  $t \notin F$  then  $error := 1$ ;
    if  $syn = -1$  then begin
      if  $pSyn > e + f + g + h$  then  $error := 1$ ;
       $syn := e + f + g + h$ ;
    end else if  $syn <> e + f + g + h$  then  $error := 1$ ;
    end else if  $t \in F$  then begin
      if  $syn = -1$  then  $pSyn := Max(pSyn, e + f + g + h)$ ;
      else if  $syn < e + f + g + h$  then  $error := 1$ ;
    end else  $error := 1$ ;
  end
end
end
end
end
end
end
end
if  $error = 1$  or  $(b \in B$  and  $syn > -1)$  then  $Result := \emptyset$ ;
else  $Result := Result \cup \{\{q'_{ok}, b\}, 1\}$ 
end
end

```

Význam premenných vo funkcii *poslednyKrok*:

- Result* - Stav po prečítaní posledného písmena
- error* - Pri simulácii kroku nastala chyba, výpočet sa musí zaseknúť
- syn* - Počet symbolov 1 v synchronizačnej postupnosti, ktorá obsahuje symbol 2
- pSyn* - Maximálny počet symbolov 1 v synchronizačnej postupnosti, ktorá neobsahuje symbol 2. Používa sa len ak hodnota premennej *syn* nie je určená ($syn = -1$)
- p, r, s, t* - Aktuálny stav
- e, f, g, h* - Čiastkový počet symbolov 1 v aktuálnej synchronizačnej postupnosti
- S* - Zoznam aktuálnych stavov po prechode na poslednom znaku

Kvôli technickej komplikovanosti dôkaz korektnosti automatu M' nebudeme uvádzať. □

5.3 Uzavretosť na označené zretáženie

Ohraničené zretáženie je operácia definovaná na jazykoch. Jej vstupy sú dva jazyky a výstupom je tretí jazyk, ktorého slová sú zretáženími slova z prvého jazyka, nového symbolu a slova z druhého jazyka. V nasledovnej leme budeme prezentovať konštrukciu, ktorá nám umožní vytvoriť takýto jazyk, ak prvý jazyk je definovaný pomocou ľubovoľného S_2^1AKA a druhý jazyk je definovaný pomocou deterministicky synchronizujúceho S_2^1AKA .

Lema 5.5. *Nech M^1 je S_2^1AKA a M^2 je deterministicky synchronizujúci S_2^1AKA . Nech c je nový symbol. Potom existuje S_2^1AKA M , ktorý akceptuje jazyk $\{w_1cw_2 \mid w_1 \in L(M^1), w_2 \in L(M^2)\}$. Navyše ak M^1 je deterministicky synchronizujúci, tak aj M je deterministicky synchronizujúci.*

Dôkaz. Nech M^1 je $(Q^1, A^1, \Sigma^1, \delta^1, q_0^1, F^1)$, M^2 je $(Q^2, A^2, \Sigma^2, \delta^2, q_0^2, F^2)$. Podľa lemy 5.4 môžeme predpokladať, že všetky akceptačné výpočty M_1, M_2 používajú len vlákna, ktoré synchronizujú.

Predpokladajme, že $Q^1 \cap Q^2 = \emptyset$.

Nech M je $(Q, A, \Sigma, \delta, q_0, F)$, kde:

$$\begin{aligned} Q &= Q^1 \cup Q^2 \cup \{q_c\} \\ A &= A^1 \cup A^2 \\ \Sigma &= \Sigma^1 \cup \Sigma^2 \cup \{c\} \\ q_0 &= q_0^1 \\ F &= F^2 \end{aligned}$$

Definujme nasledovné zobrazenie:

$$\begin{aligned} f &: Q^1 \cup (Q^1 \times \{1\}) \cup (F^1 \times \{2\}) \rightarrow Q^1 \cup (Q^1 \times \{1\}) \cup \{q_c\} \\ f(q) &= \begin{cases} q_c & q \in (F^1 \times \{2\}) \\ q & \text{inak} \end{cases} \end{aligned}$$

Nech f' je zobrazenie, ktoré aplikuje funkciu f na všetky prvky množiny. Teraz definujme δ funkciu

$$\delta(q, a) = \begin{cases} f'(\delta^1(q, a)) & a \in \Sigma_\varepsilon^1, \quad q \in Q^1 \cup (Q^1 \times \{1\}) \\ q_0^2 & a = c, \quad q = q_c \\ \delta^2(q, a) & a \in \Sigma_\varepsilon^2, \quad q \in Q^2 \cup (Q^2 \times \{1\}) \end{cases}$$

$$\{w_1cw_2 \mid w_1 \in L(M^1), w_2 \in L(M^2)\} \subset L(M)$$

Nech $w_1cw_2, w_1 \in L(M^1), w_2 \in L(M^2)$ je ľubovoľné slovo. Ukážme, že existuje akceptačný výpočet stroja M na w_1cw_2 .

1. $w_1 \in L(M^1) \wedge M^1$ je v normálnom tvare \rightsquigarrow Existuje akceptačný výpočet $T_{w_1}^{M^1}$, ktorého každý list je označený konfiguráciou $([q_f, 2]\varepsilon), q_f \in F^1$.
2. $w_2 \in L(M^2) \rightsquigarrow$ Existuje akceptačný výpočet $T_{w_2}^{M^2}$.

3. Nech $T_{w_1cw_2}^M$ je strom, ktorý dostaneme napojením $T_{w_2}^{M^2}$ na každý list $T_{w_1}^{M^1}$. Pozri obrázok 5.2. Vrcholy $p, \dots, r_1, r_2, \dots, r_n, s_1, s_2, \dots, s_n$ sú z $T_{w_1}^{M^1}$, vrcholy $t_1, \dots, t_n, \dots, u_1^1, u_2^1, \dots, u_m^n$ z $T_{w_2}^{M^2}$. Nech prvá skupina vrcholov tvorí množinu S a druhá skupina vrcholov tvorí množinu T .

Ak vrchol v z S je na výpočte $T_{w_1}^{M^1}$ označený konfiguráciou (q, x) , tak príslušný vrchol v $T_{w_1cw_2}^M$ bude označený konfiguráciou $(f(q), xcw_2)$.

Vrcholy z T budú na $T_{w_1cw_2}^M$ označené rovnakou konfiguráciou ako na $T_{w_2}^{M^2}$.

4. Ľahké vidieť

$$\forall u, v \in S, u \text{ je priamy nasledovník } v \rightsquigarrow c(v) \vdash_M c(u)$$

5. Všetky vrcholy z S označené univerzálnou konfiguráciou majú rovnakých nasledovníkov ako v úplnom strome konfigurácii. Dôkaz sporom. Predpokladajme, že $\exists v \in S, \exists c_1 : c(v) \vdash_M c_1$, v na $T_{w_1cw_2}^M$ nemá priameho nasledovníka označeného konfiguráciou c_1 . Nech $Q_1^1 = Q^1 \cup Q^1 \times \{1\}$.

$$(a) v \in S \wedge sc(v) \in A \wedge q_c \notin A \rightsquigarrow sc(v) \in Q^1$$

$$(b) (a) \wedge A \cap Q_1^1 = A^1 \rightsquigarrow sc(v) \in A^1$$

$$(c) (a) \rightsquigarrow \forall a \in \Sigma_\varepsilon(\delta(sc(v), a) = f'(\delta^1(sc(v), a)))$$

- (d) (b) \wedge (c) \wedge Konštrukcia $T_{w_1cw_2}^M \rightsquigarrow$ Na akceptačnom výpočte $T_{w_1}^{M^1}$ existuje vrchol, ktorý je označený univerzálnou konfiguráciou a výpočet neobsahuje všetkých jeho priamych nasledovníkov z úplného stromu konfigurácii. SPOR

Pomocou rovnakého postupu možno odvodiť, že všetky vrcholy z S označené existenčnou konfiguráciou majú najviac jedného priameho nasledovníka z úplného stromu konfigurácii.

6. (1.) \wedge konštrukcia $T_{w_1cw_2}^M \rightsquigarrow SP(s_1) = SP(s_2) = \dots = SP(s_n)$.
7. Definícia δ funkcie \rightsquigarrow Existuje prechod medzi vrcholmi s_i a t_i .

8. Ľahké vidieť

$$\forall u, v \in T, u \text{ je priamy nasledovník } v \rightsquigarrow c(v) \vdash_M c(u)$$

9. Všetky vrcholy z T označené univerzálnou konfiguráciou majú rovnakých nasledovníkov ako v úplnom strome konfigurácii a všetky vrcholy z T označené existenčnou konfiguráciou majú navyiac jedného priameho nasledovníka z úplného stromu konfigurácii. Rovnaká myšlienka ako pri (5.)

10. (2.) \rightsquigarrow Na cestách od vrcholov t_i do u_j^i sa nachádzajú rovnaké postupnosti synchronizačných symbolov.

11. (4.) \wedge (7.) \wedge (8.) \rightsquigarrow

$$\forall u, v \in T_{w_1cw_2}^M, u \text{ je priamy nasledovník } v \rightsquigarrow c(v) \vdash_M c(u)$$

12. (11.) \wedge koreň $T_{w_1cw_2}^M$ je označený počiatočnou konfiguráciou $\rightsquigarrow T_{w_1cw_2}^M$ je podstrom úplného stromu konfigurácii.

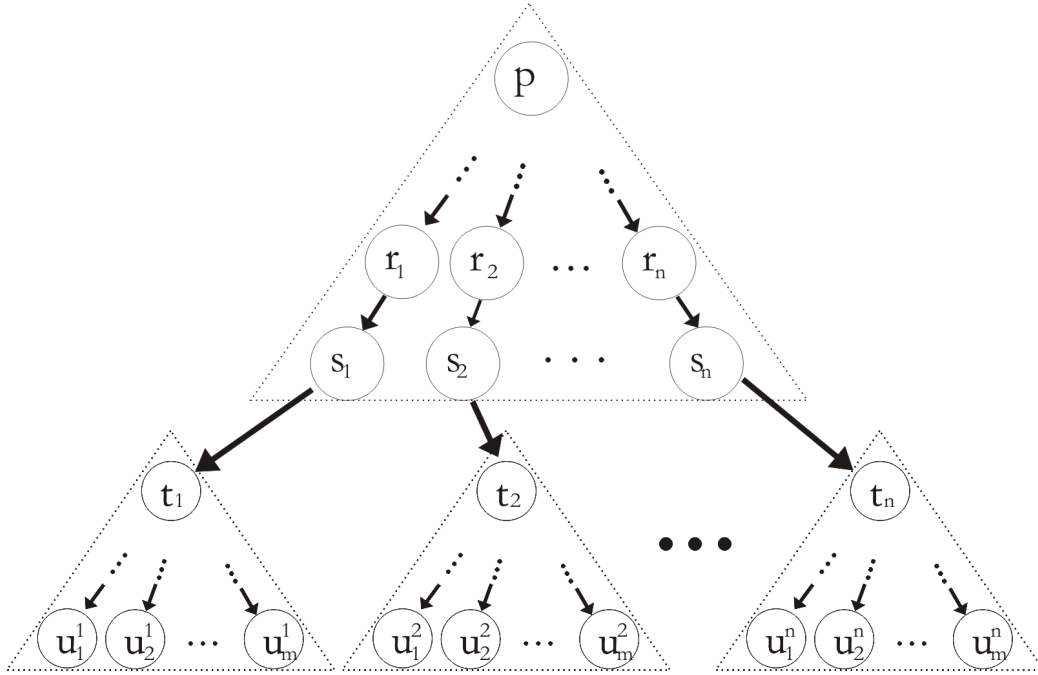
13. (5.) \wedge (9.) $\wedge q_c \notin A \rightsquigarrow$ Všetky vrcholy označené univerzálnou konfiguráciou majú rovnakých nasledovníkov, ako príslušné vrcholy v úplnom strome konfigurácii a všetky vrcholy označené existenčnou konfiguráciou majú najviac jedného priameho nasledovníka z nasledovníkov v úplnom strome konfigurácii stroja M na w_1cw_2 .

14. (6.) \wedge (10.) $\rightsquigarrow SP(u_1^1) = SP(u_2^1) = SP(u_m^n)$.

Iné uzly sú označené synchronizačnou postupnosťou tvorenou len symbolmi 1 a preto je vždy jedna postupnosť podpostupnosťou druhej.

15. (2.) $\wedge F = F^2 \rightsquigarrow$ Každý list $T_{w_1cw_2}^M$ je označený akceptačnou konfiguráciou.

16. (12.) \wedge (13.) \wedge (14.) \wedge (15.) $\rightsquigarrow T_{w_1cw_2}^M$ je akceptačný výpočet stroja M na slove $w_1cw_2 \rightsquigarrow w_1cw_2 \in L(M)$



Obrázok 5.2: Výpočet S_2^1AKA pre označené zretáženie

$$L(M) \subset \{w_1cw_2 \mid w_1 \in L(M^1), w_2 \in L(M^2)\}$$

Nech $w \in L(M)$. Ukážme, že potom $w = w_1cw_2$, kde $w_1 \in L(M^1), w_2 \in L(M^2)$.

Nech $X \subset Q$, potom označme množiny $X \cup (X \times \{1\})$, $X \cup (X \times \{2\})$, $X \cup (X \times \{1, 2\})$ ako $X_1, X_2, X_{1,2}$

1. $w \in L(M) \rightsquigarrow$ Existuje akceptačný výpočet T_w^M stroja M na vstupnom slove w .
2. Z nasledovných vlastností stroja M :
 - (a) $q_0^1 \in Q_1^1$ je počiatkový stav
 - (b) Akceptačný stav je z $F_2^2 \subset Q_{1,2}^2$
 - (c) Jediný prechod zo stavu z Q_1^1 do stavu, ktorý nepatrí do množiny Q_1^1 , je prechod do stavu q_c
 - (d) Zo stavu q_c sa dá dostať len do stavu $q_0^2 \in Q_{1,2}^2$ na symbol c
 - (e) Zo stavu z $Q_{1,2}^2$ sa dá dostať len do stavu z množiny $Q_{1,2}^2$
 - (f) Množiny $Q_1^1, \{q_c\}$ a $Q_{1,2}^2$ sú disjunktné,

vyplýva, že konfigurácie nachádzajúce sa na ľubovoľnom vlákne výpočtu T_w^M môžeme rozdeliť na tri bloky:

$$\overbrace{(q_0, w_1cw_2), \dots, (p, acw_2)}^{1.blok}, \quad \overbrace{(q_c, cw_2)}^{2.blok}, \quad \overbrace{(q_0^2, w_2), \dots, (q_f, \varepsilon)}^{3.blok},$$

kde $q_f \in F_2^2, a \in \Sigma_\varepsilon, p \in Q_1^1$. V prvom bloku sú len konfigurácie so stavmi z Q_1^1 a v treťom bloku sú len konfigurácie so stavmi z $Q_{1,2}^2$.

Označme $T_{w_1}^{M^1}$ časť výpočtu T_w^M , kde z každého vlákna zoberieme len vrcholy, ktorých konfigurácie patria do prvých dvoch blokov. Označme všetky listy $T_{w_1}^{M^1}$ symbolmi s_1, s_2, \dots, s_n .

Nech s_i je ľubovoľný z s_1, s_2, \dots, s_n . Označme $T_{w_2}^{M^2}$ časť výpočtu T_w^M , ktorá obsahuje vrcholy, ktorých predok je s_i . Je zrejmé, že vrcholy z $T_{w_2}^{M^2}$ obsahujú konfigurácie so stavom z $Q_{1,2}^2$.

3. V δ existuje prechod zo stavu p do stavu q_c na znak a len ak v δ^1 je prechod zo stavu p do stavu z $F^1 \times \{2\}$ na znak a . Navyše v δ existuje prechod zo stavu z Q_1^1 do stavu z Q_1^1 , len ak existuje rovnaký prechod aj vo funkcii δ^1 .
4. Nech $T_{w_1}^{M^1}$ je strom, ktorý dostaneme z $T_{w_1}^{M^1}$ tak, že konfigurácie na označeniach vrcholov zmeníme nasledovne. Zvyšok vstupného slova rcw_2 zmeníme na r . Stav q_c zmeníme za odpovedajúci stav z $F^1 \times \{2\}$ (podľa (3.)), iné stavy necháme nezmenené. Je zrejmé, že platí:

$$\forall u, v \in T_{w_1}^{M^1}, u \text{ je priamy nasledovník } v \rightsquigarrow c(v) \vdash_{M^1} c(u)$$

5. Každý vrchol v z $T_{w_1}^{M^1}$, ktorý je označený univerzálnou konfiguráciou má rovnakých nasledovníkov ako v úplnom strome konfigurácii stroja M^1 na slove w_1 . Dôkaz urobíme sporom.

Nech nemá nasledovníka označeného konfiguráciou c takou, že $c(v) \vdash_{M^1} c$.

- (a) Vrchol označený konfiguráciou so stavom z $F^1 \times \{2\}$ v úplnom strome konfigurácii nemá žiadneho nasledovníka \rightsquigarrow stav konfigurácie c je z Q_1^1
- (b) (a) $\rightsquigarrow c(v) \vdash_M c$, prípadne do konfigurácie, ktorú dostaneme z c zmenením stavu z $F^1 \times \{2\}$ na stav q_c

- (c) (b) \wedge konštrukcia $T_{w_1}^{M^1} \wedge A^1 \subset A \rightsquigarrow T_w^M$ obsahuje vrchol, ktorý je označený univerzálnou konfiguráciou a nemá rovnakých priamych nasledovníkov ako v úplnom strome konfigurácii. SPOR

Rovnakým spôsobom možno ukázať, že každý vrchol z $T_{w_1}^{M^1}$, ktorý je označený existenčnou konfiguráciou má najviac jedného priameho nasledovníka.

6. Je zrejmé, že pre $T_{w_1}^{M^1}$ platí, že koreň je označený počiatočnou konfiguráciou a každý list je označený akceptačnou konfiguráciou stroja M^1 na vstupnom slove w_1 .
7. Každý prechod podľa funkcie δ zo stavu z Q_1^2 do stavu z $Q_{1,2}^2$ je možný aj vo funkcii δ^2 . Preto platí:

$$\forall u, v \in T_{w_2}^{M^2}, u \text{ je priamy nasledovník } v \rightsquigarrow c(v) \vdash_{M^2} c(u)$$

8. Každý vrchol z $T_{w_2}^{M^2}$ označený univerzálnou konfiguráciou má rovnakých nasledovníkov ako v úplnom strome konfigurácii stroja M^2 na slove w_2 a každý vrchol z $T_{w_2}^{M^2}$ označený existenčnou konfiguráciou má rovnakých nasledovníkov ako v úplnom strome konfigurácii stroja M^2 na slove w_2 . Dôkaz je ekvivalentný k (5.)
9. Je zrejmé že pre $T_{w_2}^{M^2}$ platí, že koreň je označený počiatočnou konfiguráciou a každý list je označený akceptačnou konfiguráciou stroja M^2 na vstupnom slove w_2 .
10. (8.) \wedge (9.) \rightsquigarrow Ak ukážeme, že každý list $T_{w_2}^{M^2}$ má rovnakú synchronizačnú postupnosť, tak $T_{w_2}^{M^2}$ je akceptačný výpočet.
11. Označme listy T_w^M , ktoré sú nasledovníkmi vrcholu s_k symbolmi $u_1^k, u_2^k, \dots, u_{m_k}^k$.

Ukážme, že $SP(s_1) = SP(s_2) = \dots = SP(s_n)$. Sporom.

- (a) Nech $SP(s_i) \neq SP(s_j)$
- (b) T_w^M je akceptačný výpočet $\rightsquigarrow SP(u_1^i) = SP(u_2^i) = \dots = SP(u_{m_i}^i) = SP(u_1^j) = SP(u_2^j) = \dots = SP(u_{m_j}^j)$
- (c) Nech funkcia g vráti počet symbolov 1 v synchronizačnej postupnosti vrcholu z T_w^M , t.j. $g(x) = \#_1(SP(x))$.
- (a) \wedge (b) $\rightsquigarrow k = g(u_1^i) - g(s_i) = g(u_2^i) - g(s_i) = \dots = g(u_{m_i}^i) - g(s_i) \neq g(u_1^j) - g(s_j) = g(u_2^j) - g(s_j) = \dots = g(u_{m_j}^j) - g(s_j) = l$, kde k, l sú vhodné prirodzené čísla.

- (d) Nech T_i, T_j sú časti výpočtu T_w^M , ktoré obsahujú vrcholy, ktorých predok je s_i a s_j . Je ľahké vidieť, že synchronizačná postupnosť listov T_i (resp. T_j) je 1^k2 (resp. 1^{l_2}).
- (e) $(8.) \wedge (d) \rightsquigarrow T_i, T_j$ sú akceptačné výpočty stroja M^2 na w_2 .
- (f) $(c) \wedge (e) \wedge M^2$ je deterministicky akceptujúci \rightsquigarrow SPOR

Dôkaz predchádzajúceho tvrdenia navyše ukázal, že listy $T_w^{M^2}$ majú rovnakú synchronizačnú postupnosť.

- 12. $(5.) \wedge (6.) \wedge (11.) \rightsquigarrow T_{w_1}^{M^1}$ je akceptačný výpočet stroja M^1 na vstupnom slove w_1 .
- $(10.) \wedge (11.) \rightsquigarrow T_{w_2}^{M^2}$ je akceptačný výpočet stroja M^2 na vstupnom slove w_2 .

O deterministickom synchronizovaní:

Ak M^1 je deterministický synchronizujúci S_2^1 AKA, tak každý jeho akceptačný výpočet na slove w musí synchronizovať rovnako. Dôkaz urobíme priamo.

Nech T_1, T_2 sú dva akceptačné výpočty stroja M na w .

Už sme ukázali, že $w = w_1cw_2$ a každý akceptačný výpočet možno rozdeliť na časť, z ktorej možno skonštruovať akceptačný výpočet M^1 na w_1 a niekoľko častí, z ktorých možno skonštruovať akceptačný výpočet M^2 na w_2 .

Prvá časť výpočtu (na slove w_1) musí synchronizovať rovnako (listy sú označené rovnakou postupnosťou symbolov 1) v T_1 aj v T_2 , inak možno skonštruovať inak synchronizujúce výpočty stroja M^1 na w_1 .

Ale aj druhá časť výpočtu musí synchronizovať rovnako v T_1 aj v T_2 , lebo inak možno skonštruovať inak synchronizujúce výpočty M^2 na w_2 . Tým je tvrdenie dokázané. □

Poznámka 5.1. *Myšlienka konštrukcie z predchádzajúcej lemy vychádza z tvrdenia:*

$$\forall a_1^1, a_2^1, \dots, a_{m_1}^1, a_1^2, a_2^2, \dots, a_{m_2}^2, \dots, a_1^n, a_2^n \dots a_{m_n}^n ($$

$$\forall x_1, y_1 \in \{a_1^1, \dots, a_{m_1}^1\}, x_2, y_2 \in \{a_1^2, \dots, a_{m_2}^2\}, \dots, x_n, y_n \in \{a_1^n, \dots, a_{m_n}^n\} ($$

$$\sum_{i=1}^n x_i = \sum_{i=1}^n y_i))$$

\Updownarrow

$$\forall i \in \mathbb{N}_n (\forall j, k \in \mathbb{N}_{m_i} (a_j^i = a_k^i))$$

a nahradenia n rovností na rôzne čísla za $m_1 \cdot m_2 \cdot \dots \cdot m_n$ rovností na rovnaké číslo.

Táto myšlienka sa dá okrem iného použiť pri konštrukcii S_2^1 AKA, pre jazyk $\{w \mid \#_a(w) = \#_b(w) \wedge \#_c(w) = \#_d(w)\}$. Každý výpočet by používal 4 procesy, kde:

1. proces synchronizuje na znaky a, c
2. proces synchronizuje na znaky a, d
3. proces synchronizuje na znaky b, c
4. proces synchronizuje na znaky b, d

Na konci výpočtu procesy synchronizujú so symbolom 2. Je ľahké vidieť, že takéto zariadenie akceptuje jazyk $\{w \mid \#_a(w) = \#_b(w) \wedge \#_c(w) = \#_d(w)\}$.

Teraz vyslovíme tvrdenie, ktoré hovorí o triede jazykov definovaných pomocou deterministicky synchronizujúcich S_2^1 AKA.

Veta 5.2. *Trieda deterministicky synchronizujúcich S_2^1 AKA je uzatvorená na označené zretazenie.*

Dôkaz. Vyplýva z lemy 5.5. □

Poznámka 5.2. *Je dobré si uvedomiť, že konštrukcia uvedená v leme 5.5 rovnako dobre funguje na všeobecné zretazenie pre jazyky, kde začiatok slova z druhého jazyka je ľahko identifikovateľný. Napríklad nám dáva návod ako možno skonštruovať S_2^1 AKA, ktorý akceptuje slová z jazyka $a^n b^n a^m b^m$.*

6 Záver

V práci sme skúmali, aký vplyv na výpočtovú silu alternujúceho zariadenia majú niektoré obmedzené používania synchronizačných symbolov.

Ukázali sme, že kardinalita abecedy synchronizačných symbolov má veľký vplyv na výpočtovú silu modelu. Jeden synchronizačný symbol nepridá žiadnu výpočtovú silu. S dvomi symbolmi dokážeme simulovať ľubovoľný počet synchronizačných symbolov, ale to len za cenu nárastu časovej zložitosti.

Definovali sme synchronizáciu s jedným synchronizačným symbolom a zarážkou a preskúmali sme konečné automaty, s takýmto spôsobom synchronizácie.

Ukázali sme, že pre tento model existuje normálny tvar, v ktorom každý proces synchronizuje v rovnakej synchronizačnej postupnosti, ktorá môže byť dlhšia ako ľubovoľná konštanta. Pri konštrukcii sme vyvinuli techniku použitia jedného koreňového procesu, ktorý simuluje procesy pôvodného zariadenia, ktoré ešte nesynchronizovali k krát. Ak nejaký proces chce synchronizovať viac ako k krát, simulácia tohto procesu pokračuje v samostatnom novom vlákne.

Najzaujímavejší výsledok hovorí, že trieda definovaná alternujúcimi konečnými automatmi synchronizujúcimi s jedným symbolom a zarážkou, ktoré pre konkrétne slovo vyprodukujú vždy rovnakú synchronizačnú postupnosť, je uzatvorená na ohraničené zreťazenie. Pri dôkaze tohto tvrdenia sme vyvinuli techniku, ktorá nám dovolila nahradiť viacej rovností s viacerými členmi na jednu rovnosť s viacerými členmi.

Ukázali sme, že trieda obsahuje jazyky, na ktorých akceptáciu nám nestačia ani bezkontextové prostriedky – jazyk $a^n b^n c^n$. Pri konštrukcii zariadenia pre tento jazyk sme prezentovali, ako v modely možno realizovať taktiku Rozdeľuj a panuj. Prvé tri procesy overili rovnosť počtu symbolov a, b, c , štvrtý proces overil tvar slova.

Otvoreným problémom zostala platnosť hypotézy, či existuje jazyk, ktorý je akceptovaný sekvenčným počítadlovým automatom, a pre ktorý neexistuje synchronizovane alternujúci konečný automat s jedným synchronizačným symbolom a zarážkou. Dôkaz alebo vyvrátenie hypotézy sa ukázalo ako veľký problém a napriek tomu, že sa ho nepodarilo vyriešiť, ukázal nám mnohé vlastnosti takto definovaného modelu. Napríklad výsledok o uzavretosti na označené zreťazenie vznikol pri pokuse ukázať, že $a^n b^n c^m d^m$ je takýto jazyk. Z našich snažení sa zdá, že vhodný jazyk, ktorý dokazuje platnosť hypotézy, je jazyk dobre uzátvorkovaných výrazov s jedným párom zátvoriek – D_1 .

Zoznam obrázkov

3.1	SAPA simulujúci P2A, prvý krok indukcie	15
3.2	SAPA simulujúci P2A, indukčný krok	16
3.3	Tvar výpočtu SAPA M' na slove	17
3.4	SAPA simulujúci P2A, akceptačný výpočet	21
5.1	Výpočet na slove $a^n b^n c^n$	32
5.2	Výpočet S_2^1 AKA pre označené zreťazenie	43

Referencie

- [1] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the ACM*, 28:114–133, 1981.
- [2] Juraĳ Hromkoviĉ, Juhani Karhumäki, Branislav Rován, and Anna Slobodová. On the power of synchronization in parallel computation. *Discrete Applied Mathematics*, 32:155–182, 1991.
- [3] Juraĳ Hromkoviĉ, Katsushi Inoue, Branislav Rován, Anna Slobodová, Itsuo Takanami, and Klaus W. Wagner. On the power of one-way synchronized alternating machines with small space. *International Journal of Foundations of Computer Science*, 3:65–79, 1992.
- [4] Branislav Rován. *Formálne Jazyky a Automaty (skriptá)*. 2007. Spísal: Michal Foríšek.
- [5] Dexter C. Kozen. *Automata and Computability*. Birkhäuser, 1997.
- [6] Branislav Rován. *Teória paralelných výpoĉtov*. 2000. Elektronické poznámky k prednáškam.