Vysoké učení technické v Brně

Fakulta informačních technologií

Projekt v rámci předmětu TJD

Formální modely jazyků založené na automatech a jejich modifikace

2005

Radek Bidlo

Abstrakt

Tento dokument popisuje základní modely formálních jazyků založené na automatech a zavádí některé další modifikace, kterými je možné zvýšit jejich akceptační schopnosti. Detailněji je pojednáno o oboustranných zásobníkových automatech nad volnou grupou, které vychází z klasických zásobníkových automatů. Na závěr je prezentován důkaz ekvivalence třídy jazyků přijímaných těmito automaty s třídou rekurzívně vyčíslitelných jazyků.

Obsah

1 Úvod

Automaty představují základní formální model jakéhokoliv výpočtu. Ten je zpravidla reprezentován určitou posloupností přechodů konkrétního automatu mezi jednotlivými jeho konfiguracemi podle přesně daných pravidel, které jsou součástí definice automatu. Výpočet pak můžeme prohlásit za *úspěšný*, nachází-li se automat po provedení posledního výpočetního kroku v konfiguraci, která je definována jako cílová. V opačném případě je výpočet neúspěšný. Cílová konfigurace bývá zpravidla reprezentována koncovým stavem automatu, ale mohou přibýt i další podmínky jako je například vyprázdnění zásobníku a podobně. Samozřejmostí je kompletní načtení vstupního řetězce.

Z hlediska teorie formálních jazyků chápeme automaty jako tzv. *akceptory jazyků*. Jazyk přijímaný daným automatem pak zpravidla definujeme jako množinu všech vět, které je schopen automat zpracovat a přitom po provedení posledního kroku skončit v některé z cílových konfigurací. Posloupnost takovýchto kroků vedoucích do některé cílové konfigurace při přijímání určitého řetězce je pak možno chápat jako úspěšný výpočet daného automatu. Nejznámější z automatů, které se během vývoje teoretické informatiky ustálily jako jakési základní verze, jsou zejména konečné automaty, zásobníkové automaty a Turingovy stroje. My se budeme podrobněji zabývat konečnými a zásobníkovými automaty.

V následujících kapitolách si uvedeme vždy přesnou definici a ukážeme, jakou třídu jazyků jsou schopny ve své základní variantě definovat. Dále zavedeme některá jejich rozšíření případně omezení a budeme studovat akceptační schopnosti nově vzniklých modelů. Podrobně se zaměříme zejména na automaty zásobníkové. U čtenáře se předpokládá základní znalost teorie formálních jazyků a algebry, viz [1] a [2].

2 Základní pojmy a definice

V této kapitole zopakujeme naprosté základy teorie formálních jazyků a definujeme si formálně pojmy jako jsou *abeceda*, *řetězec*, a *jazyk nad abecedou*. Základní definice jsou převzaty z [3].

2.1 Symbol, abeceda, řetězec

Definice 2.1 *Abecedou* nazveme každou konečnou neprázdnou množinu prvků, které nazáváme *symboly* této abecedy.

Libovolná sekvence symbolů určité abecedy tvoří řetězec. Označíme-li symbolem ε prázdný řetězec, tedy řetězec, který neobsahuje žádné symboly, můžeme všechny řetězce definovat rekurzívně.

Definice 2.2 Nechť Σ je libovolná abeceda.

1. ε je řetězec nad Σ
2. je-li x řetězec nad Σ a $a \in \Sigma$, pak také xa je řetězec nad Σ

2.2 Jazyk nad abecedou

Uvažujme libovolnou abecedu Σ. Označme Σ^* množinu všech řetězců nad abecedou Σ včetně řetězce prázdného a Σ^+ množinu všech řetězců nad abecedou Σ vyjma řetězce prázdného, tedy $\Sigma^+ = \Sigma^* - \{\varepsilon\}$. Jinými slovy, Σ^+ obsahuje všechny *neprázdné* řetězce nad abecedou Σ. Σ^* resp. Σ^+ se nazývá *iterace* resp. *pozitivní iterace* množiny Σ. Definujme nyní jazyk nad abecedou Σ.

Definice 2.3 Nechť Σ je abeceda a nechť $L \subseteq \Sigma^*$. Potom L je *jazyk* nad abecedou Σ.

Z této definice je zřejmé, že jazykem nad určitou abecedou může být vlastně libovolná podmnožina iterace této množiny. Otázkou zůstává, jak takový jazyk popsat. Možností je několik. Jedna z nejméně praktických je přímý výčet prvků jazyka. Tato varianta je však zcela nepoužitelná v případě nekonečných jazyků a problémy nastávají již při pokusu o popis rozsáhlých konečných jazyků. Proto byly vyvinuty určité formální modely, které poskytují konečné prostředky pro popis obecně nekonečných jazyků. Tyto modely můžeme rozdělit do dvou základních skupin — gramatiky a automaty. My se v tomto projektu zaměříme právě na automaty.

2.3 Chomského hierarchie jazyků

V roce 1956 rozdělil americký jazykovědec Avram Noam Chomsky jazyky do hierarchie podle tvarů přepisovacích pravidel gramatik, kterými mohou být generovány. Tato hierarchie byla jedním z nejvýznamnějších objevů dvacátého století v oblasti teorie formálních jazyků a dosud nese jeho jméno. Přestože se postupem času objevily další formální modely, které svými vyjadřovacími schopnostmi zasahují přes několik tříd jazyků Chomského hierarchie (tedy jsou schopny popsat z každé skupiny pouze některé jazyky), jedná se stále o jedno ze základních členění jazyků a třídy jazyků této hierarchie bývají srovnávány v mnoha důkazech ekvivalence tříd jazyků definovaných jinými formálními modely. Protože je tento dokument zaměřen hlavně na automaty jako prostředky pro popis formálních jazyků, popišme si jednotlivé třídy jazyků Chomského hierarchie pouze neformálně.

- jazyky typu 0 — zahrnují všechny jazyky s gramatickým základem; všechny tyto jazyky jsou přijímány Turingovými stroji a jsou známé taktéž pod označením *rekurzívně vyčíslitelné jazyky*
- jazyky typu 1 — známé též pod označením *kontextové jazyky*; všechny tyto jazyky jsou přijímány lineárně ohraničenými nedeterministickými Turingovými stroji
- jazyky typu 2 — označovány též jako *bezkontextové jazyky*; tyto jazyky jsou přijímány nedeterministickými zásobníkovými automaty (viz dále) a představují určitý teoretický základ syntaxe mnoha programovacích jazyků
- jazyky typu 3 — označovány většinou jako *regulární jazyky*, které jsou přijímány *konečnými automaty* (viz dále)

Jednotlivé třídy jsou vzájemně vázány ostrou inkluzí. Označíme-li symboly **RE**, **CS**, **CF** a **REG** postupně třídu rekurzívně vyčíslitelných, kontextových, bezkontextových a regulárních jazyků, potom platí **REG**$\subset$**CF**$\subset$**CS**$\subset$**RE**.

2.4 Grupy a volné grupy

V této části uvedeme velmi stručně definici grupy a její zobecněné varianty — volné grupy (zejména s ohledem na naše další použití v kapitole 5). Teorie grup tvoří samostatnou algebraickou disciplínu a je velmi obsáhlá. My si zde proto uvedeme opravdu jen to nejdůležitější, co budeme v dalších částech tohoto dokumentu potřebovat.

Definice 2.4 Nechť V je množina. Strukturu $(V, \cdot, e)$ nazveme *grupou*, jestliže

- $\cdot : V \times V \to V$ je binární asociativní operátor (uzavřený na V)
- existuje unikátní prvek $e \in V$ takový, že $e \cdot a = a \cdot e = a$ pro každé $a \in V$
- pro každé $a \in V$ existuje $\overline{a} \in V$ takové, že $a \cdot \overline{a} = \overline{a} \cdot a = e$;

Definice 2.5 Nechť V je abeceda a nechť $\cdot$ je binární asociativní operátor konkatenace. Strukturu $(V^\circ, \cdot, \varepsilon)$ nazveme *volnou grupou* generovanou množinou V a operací konkatenace, jestliže

- pro každé dva řetězce $x, y \in V^\circ$ platí $x \cdot y \in V^\circ$
- existuje unikátní řetězec ε zvaný prázdný, který je neutrálním prvkem této struktury (tj. pro každý řetězec $x \in V^\circ$ platí $x \cdot \varepsilon = \varepsilon \cdot x = x$)
- pro každý řetězec tvaru $x = x_1x_2 \dots x_n \in V^\circ$, $n \geq 0$ existuje $\overline{x} = \overline{x_n} \dots \overline{x_2}\overline{x_1}$ zvaný *inverzní* takový, že platí $x \cdot \overline{x} = \overline{x} \cdot x = \varepsilon$

3 Konečné a zásobníkové automaty

Automaty, chápané jako formální modely výpočtu nebo jako akceptory jazyků, lze rozdělit do následujících tříd.

1. Konečné automaty
2. Zásobníkové automaty
3. Turingovy stroje

My se v tomto dokumentu zaměříme na konečné a zásobníkové automaty. V této kapitole si uvedeme jejich základní definici. V další kapitole potom ukážeme, jak vhodným rozšířením základních modelů můžeme výrazně zvýšit jejich akceptační schopnosti. Poznamenejme, že dále se již automaty nebudeme zabývat ve smyslu modelů výpočtu, ale ve smyslu akceptorů jazyků.

3.1 Konečné automaty

Konečný automat je nejslabší formální model z výše uvedené skupiny. Jak jsme již uvedli výše, konečné automaty definují pouze třídu regulárních jazyků. Uveďme si nyní jeho formální definici.

Definice 3.1 *Konečný automat* je pětice, $M = (Q, \Sigma, R, q_0, F)$, kde

- Q je konečná množina vnitřních stavů
- Σ je konečná vstupní abeceda
- R je konečná množina pravidel tvaru $pa \to q$, kde $p, q \in Q$, $a \in \Sigma$; v případě, že platí $a \in \Sigma \cup \{\varepsilon\}$, nazývá se konečný automat *rozšířený*; pokud navíc pro každý symbol $a \in \Sigma$ platí, že existuje pro každý stav $q \in Q$ nejvýše jedno pravidlo $qa \to p$, pro nějaké $p \in Q$, nazývá se takovýto konečný automat *deterministický*
- $q_0 \in Q$ je počáteční stav
- $F \subseteq Q$ je množina koncových stavů

Konfigurací konečného automatu M rozumíme řetězec qx, kde $q \in Q$ a $x \in \Sigma^*$. Pokud qax a px jsou dvě konfigurace a $qa \to p \in R$, pak automat M provádí *přechod* z konfigurace qax do konfigurace px podle pravidla $qa \to p$ a píšeme

$$qax \vdash px[qa \to p]$$

nebo stručněji $qax \vdash px$. Symbol $\vdash$ označuje relaci přechodu mezi jednotlivými konfiguracemi. Symboly $\vdash^n$, $\vdash^+$ a $\vdash^*$ označují postupně posloupnost přechodů délky n, $n \geq 0$, tranzitivní uzávěr a reflexivní-tranzitivní uzávěr relace přechodu $\vdash$.

Jazyk přijímaný automatem M je definován jako

$$L(M) = \{w | w \in \Sigma^* \wedge q_0 w \vdash^* f \wedge f \in F\}$$

Poznamenejme na závěr tohoto odstavce, že deterministické konečné automaty mají stejnou vyjadřovací schopnost jako nedeterministické, tedy definují rovněž třídu regulárních jazyků. Členění z tohoto hlediska však pro nás není podstatné a proto se jím nebudeme zabývat.

3.2 Zásobníkové automaty

Zásobníkový automat (pushdown automaton, PDA) svým způsobem představuje určité rozšíření konečného automatu v tom smyslu, že obsahuje zásobník jako vnější paměť teoreticky neomezené velikosti. Jeho generativní síla je v Chomského hierarchii o stupeň výše — jak jsme již uvedli dříve, *nedeterministické* zásobníkové automaty definují třídu bezkontextových jazyků. Na rozdíl od konečných automatů, deterministické zásobníkové automaty mají nižší vyjadřovací schopnost než nedeterministické. Označíme-li $\mathbf{PDA}(NED)$ třídu jazyků přijímaných nedeterministickými zásobníkovými automaty a $\mathbf{PDA}(DET)$ třídu jazyků přijímaných deterministickými zásobníkovými automaty, pak platí

$$\mathbf{REG} \subset \mathbf{PDA}(DET) \subset \mathbf{PDA}(NED) = \mathbf{CF}$$

Pokud budeme v dalších částech hovořit o zásobníkovém automatu (nebo jeho modifikaci), budeme mít vždy na mysli nedeterministickou variantu. Uveďme si nyní formální definici.

Definice 3.2 *Zásobníkový automat* je n-tice $M = (Q, \Sigma_I, \Sigma_{PD}, R, Z, q_0, F)$, kde

- Q je konečná množina stavů
- Σ_I je konečná vstupní abeceda
- Σ_{PD} je konečná zásobníková abeceda
- R je konečná množina pravidel tvaru $Apa \to wq$, kde $A \in \Sigma_{PD}$, $p, q \in Q$, $a \in \Sigma_I \cup \{\varepsilon\}$, and $w \in \Sigma_{PD}^*$
- $Z \in \Sigma_{PD}$ je počáteční symbol zásobníku
- $q_0 \in Q$ je počáteční stav
- $F \subseteq Q$ je množina koncových stavů

Konfigurací zásobníkového automatu M rozumíme řetězec yqx, kde $q \in Q$, $y \in \Sigma_{PD}^*$ a $x \in \Sigma_I^*$. Pokud $uAqav$ a $uwpv$ jsou dvě konfigurace a $Aqa \to wp \in R$, pak automat M provádí *přechod* z konfigurace $uAqav$ do konfigurace $uwpv$ podle pravidla $Aqa \to wp$ a píšeme

$$uAqav \vdash uwpv[Aqa \to wp]$$

nebo stručněji $uAqav \vdash uwpv$. Symboly $\vdash^n$, $\vdash^+$ a $\vdash^*$ označují postupně posloupnost přechodů délky n, $n \geq 0$, tranzitivní uzávěr a reflexivní-tranzitivní uzávěr relace přechodu $\vdash$.

Jazyk přijímaný automatem M může být definován třemi způsoby:

a) přechodem do koncového stavu: $L(M_f) = \{w | w \in \Sigma_I^* \wedge Zq_0w \vdash^* rf$, where $f \in F, r \in \Sigma_{PD}^*\}$

b) s vyprázdněním zásobníku: $L(M_e) = \{w | w \in \Sigma_I^* \wedge Zq_0w \vdash^* p$, where $p \in Q\}$

c) přechodem do koncového stavu a s vyprázdněním zásobníku: $L(M_{fe}) = \{w | w \in \Sigma_I^* \wedge Zq_0w \vdash^* f$, where $f \in F\}$

Klasické konečné a zásobníkové automaty jsou v teorii formálních jazyků všeobecně známé a tvoří významný formální základ pro nesčetné množství praktických aplikací (zejména v oblasti lexikální a syntaktické analýzy). V dalších částech si ale ukážeme některé další prostředky, jakými můžeme schopnosti konečných a zásobníkových automatů vylepšit.

4 Konečné automaty nad grupami

Náplň této kapitoly je čerpána z [4] a [5].

Nechť $\mathbf{K} = (M, \circ, e)$ je grupa s bázovou množinou M, binární operací $\circ$ a neutrálním prvkem e.

Definice 4.1 *Rozšířený konečný automat* (extended finite automaton, EFA) nad grupou $\mathbf{K}$ je šestice $A = (Q, \Sigma, \mathbf{K}, R, q_0, F)$, kde

- Q, Σ, q_0, F mají stejný význam jako v případě běžného konečného automatu, tedy konečná množina stavů, konečná vstupní abeceda, počáteční stav a množina koncových stavů (v tomto pořadí)
- $R : Q \times (\Sigma \cup \{\varepsilon\}) \to P_f(Q \times M)$

Takto definovaný automat můžeme považovat za běžný konečný automat, který má navíc registr pro uchování libovolného prvku z M. Relace $(q, m) \in R(s, a)$, kde $q, s \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $m \in M$ znamená, že automat A změní svůj současný stav s na stav q, přečte ze vstupní pásky symbol a a do registru uloží hodnotu $x \circ m$, kde x je starý obsah registru. Startovací hodnota registru je e.

Konfigurací takovéhoto automatu rozumíme trojici (q, u, m), kde $q \in Q$, $u \in \Sigma^*$ a $m \in M$. Nechť potom (q, aw, m) a $(s, w, m \circ r)$ jsou dvě konfigurace, $r \in M$, a nechť $(s, r) \in R(q, a)$. Pak píšeme

$$(q, aw, m) \vdash (s, w, m \circ r)$$

a říkáme, že automat A provádí krok (přechod) z konfigurace (q, aw, m) do konfigurace $(s, w, m \circ r)$ podle $(s, r) \in R(q, a)$.

My se dále zaměříme na rozšířené konečné automaty nad volnými grupami. Připomeňme, že pro každou (ne-abelovskou) grupu $\mathbf{K}$ existuje homomorfismus z volné grupy do $\mathbf{K}$. Volnou grupu generovanou nějakou neprázdnou spočetnou množinou M označíme $\mathbf{F}(M)$. Grupu s n generátory potom označíme jako $\mathbf{F}_n$. Dále označme $\mathbf{L}(EFA(\mathbf{F}_n))$ třídu jazyků přijímaných rozšířenými konečnými automaty nad volnými grupami s n generátory.

Připomeňme z [7] definici aditivní valenční gramatiky (additive valence grammar), která generuje jazyk podobným způsobem jako je definován jazyk přijímaný výše popsaným rozšířeným konečným automatem. Aditivní valenční gramatika je pětice, $G = (N, T, P, S, v)$, kde (N, T, P, S) je běžná gramatika Chomského hierarchie a v je zobrazení z P do množiny celých čísel. Jazyk generovaný gramatikou G je tvořen všemi řetězce nad terminální abecedou T takovými, které byly vygenerovány posloupností pravidel p_1, p_2,..., $p_n \in P$ a zároveň $v(p_1) + v(p_2) + \ldots + v(p_n) = 0$. Protože $\mathbf{F}_1$ a aditivní grupa celých čísel jsou isomorfní, potom každý jazyk generovaný nějakou regulární gramatikou s aditivní valencí je obsažen v $\mathbf{L}(EFA(\mathbf{F}_1))$ a naopak.

4.1 Nová charakterizace bezkontextových jazyků

V předchozích kapitolách jsme si již definovali vše potřebné pro to, abychom mohli uvést významný výsledek této kapitoly převzatý z [5].

Věta 4.1 $\mathbf{CF} = \mathbf{L}(EFA(\mathbf{F}_2))$

Jinými slovy tento matematický zápis říká, že každý bezkontextový jazyk může být přijímán rozšířeným zásobníkovým automatem nad volnou grupou se dvěma generátory. Uveďme si alespoň nástin důkazu této věty, který bude sestávat ze dvou částí.

První část dokazuje inkluzi $\mathbf{CF} \subseteq \mathbf{L}(EFA(\mathbf{F}_2))$, tedy skutečnost, že třída bezkontextových jazyků je obsažena v třídě jazyků přijímaných rozšířenými konečnými automaty nad volnými grupami s dvěma generátory.

Proof

Nechť L je bezkontextový jazyk přijímaný nějakým zásobníkovým automatem $P = (Q, V, \Delta, R, Z, q_0, F)$ koncovým stavem a vyprázdněním zásobníku, kde $\Delta = \{X_1, X_2, \ldots, X_n\}$, $X_1 = Z$. Bez ztráty obecnosti dále můžeme předpokládat, že pro libovolný koncový stav již není definován žádný přechod. Položme $M = \{c_1, c_2, \ldots, c_n\}$ a zaveďmě zobrazení $\sigma : \Delta \to \mathbf{F}(M)$ jako $\sigma(X_i) = c_i$, $1 \leq n$. Nyní konstruujme rozšířený konečný automat nad volnou grupou $\mathbf{F}(M)$, $A = (Q \cup \{s_0\}, V, \mathbf{F}(M), f, s_0, F)$, kde

$$f(s_0, \varepsilon) = \{(q_0, c_1)\}$$

a

$$f(q, a) = \bigcup_{X \in \Delta} \{(p, (\sigma(X))^{-1}\sigma(Y_m) \ldots \sigma(Y_2)\sigma(Y_1)) | Xqa \to Y_1 Y_2 \ldots Y_m p \in R\} \cup$$

$$\bigcup_{X \in \Delta} \{(p, (\sigma(X))^{-1} | Xqa \to p \in R\}$$

pro všechna $a \in V \cup \{\varepsilon\}$ a $q \in Q$. Matematickou indukcí pak lze ukázat, že $R_1 R_2 \ldots R_m qxy \vdash^n R'_1 R'_2 \ldots R'_s q' y$ v P právě když

$$(q, xy, \sigma(R_m)\sigma(R_{m-1}) \ldots \sigma(R_1)) \vdash^n (q', y, \sigma(R'_s)\sigma(R'_{s-1}) \ldots \sigma(R'_1))$$

v A. Oba automaty pak jsou v každém kroku ve stejném stavu a neutrální prvek volné grupy se v registru automatu A objeví právě v okamžiku, když se vyprázdní zásobník automatu P. Jazyky generované automatem A a automatem PDA jsou tedy shodné. Protože každá volná grupa je zároveň podgrupou binární volné grupy, je důkaz hotový. □

Důkaz inkluze $\mathbf{L}(EFA(\mathbf{F}_2)) \subseteq \mathbf{CF}$ uveďme již pouze stručně. V tomto případě budeme na počátku uvažovat libovolný rozšířený konečný automat nad volnou grupou s dvěma generátory. Po konstrukci vhodné pravé lineární gramatiky dále definujeme tzv. *směšovací* operaci (shuffle operation) $\wr$ rekurzívně takto:

$$(u \wr \varepsilon) = (\varepsilon \wr u) = \{u\}$$

a

$$(au \wr bw) = a(u \wr bv) \cup b(au \wr v)$$

kde u, v jsou řetězce a a, b symboly. Přirozené rozšíření této operace na jazyky provedeme jako

$$L_1 \wr L_2 = \bigcup_{u \in L_1, v \in L_2} (u \wr v)$$

S využitím Dykova jazyka (Dyck language) řádu 2 (viz strana 603 v [3]) a vlastností uzávěru rodiny bezkontextových jazyků dokážeme, že $L(A)$ je skutečně bezkontextový. Případné zájemce odkazujeme na [5], kde je důkaz kompletní. ■

Již bez důkazu si uveďme jakýsi souhrn této části, který je převzat rovněž z [5].

Věta 4.2 $REG = \mathbf{L}(EFA(\mathbf{F}_0)) \subset \mathbf{L}(EFA(\mathbf{F}_1)) \subset \mathbf{L}(EFA(\mathbf{F}_2)) = \mathbf{CF}$

Zmiňme se stručně i o deterministických konečných automatech nad volnými grupami. Je všeobecně známo, že běžné deterministické a nedeterministické konečné automaty mají stejnou vyjadřovací schopnost. Z tohoto pohledu je však překvapující, že deterministické konečné automaty nad volnými grupami mají menší vyjadřovací sílu než jejich nedeterministické varianty. Podívame-li se však na tuto problematiku ze strany bezkontextových jazyků, jejichž formálními modely jsou i zásobníkové automaty, jedná se vlastně o přirozenou vlastnost, neboť deterministické zásobníkové automaty jsou taktéž slabší než jejich nedeterministické varianty. Více o této problematice lze opět najít v [5].

4.2 Diskuse ke konečným automatům

V této kapitole jsme si demonstrovali některé další modifikace běžných konečných automatů. Vidíme, že jejich vhodným rozšířením jsme schopni do určité míry zvýšit jejich vyjadřovací schopnosti. Nejsme tedy omezeni pouze třídou regulárních jazyků.

V následující kapitole se budeme pro změnu zabývat zásobníkovými automaty a ukážeme, že podobný přístup je aplikovatelný i v této oblasti.

5 Oboustranné zásobníkové automaty nad volnými grupami

Množina všech řetězců nad určitou abecedou je obvykle vyjádřena volným monoidem generovaným symboly této abecedy a operací konkatenace. Neutrálním prvkem je pak prázdný řetězec ε. Nad množinou všech řetězců jsou definovány relace přímé derivace a další související operace. My však volné monoidy částečně opustíme. Každý prvek původní abecedy rozšíříme o jeho inverzní variantu a množinu všech řetězců nad nově vzniklou abecedou budeme definovat pomocí volné grupy generované touto abecedou a operací konkatenace. Kromě neutrálního prvku ε získáme i inverzní řetězce. Konkatenací každého řetězce s jeho inverzní variantou pak získáme právě prázdný řetězec ε (viz definice 2.4 a 2.5). Tato kapitola je ispirována článkem [6]. Zavádí však zcela nový přístup, zjednodušuje konstrukci pravidel automatu a zprůhledňuje důkaz.

5.1 Frontové gramatiky

Pro definici a konstrukci oboustranných zásobníkových automatů nad volnými grupami využijeme tzv. *frontové gramatiky* (viz též [8]).

Definice 5.1 *Frontová gramatika* je šestice $G = (V, T, W, F, s, P)$, kde

- V a W jsou dvě konečné abecedy, pro které platí $V \cap W = \varnothing$
- $T \subseteq V$, $F \subseteq W$
- $s \in (V - T)(W - F)$ je axiom
- $P \subseteq V \times (W - F) \times V^* \times W$ je konečná relace taková, že pro každé $a \in V$ existuje prvek $(a, b, x, c) \in P$.

Jestliže $u, v \in V^*W$ jsou řetězce tvaru $u = arb$, $v = rxc$, kde $a \in V$, $r, x \in V^*$, $b, c \in W$ a $(a, b, x, c) \in P$, pak říkáme, že u přímo derivuje v ve frontové gramatice G podle pravidla (a, b, x, c). Píšeme

$$arb \Rightarrow rxc[(a, b, x, c)]$$

Symboly $\Rightarrow^n$, $\Rightarrow^*$ a $\Rightarrow^+$ označují postupně derivaci délky n, $n \geq 0$, tranzitivní uzávěr a reflexivní-tranzitivní uzávěr relace přímé derivace $\vdash$.

Jazyk generovaný frontovou gramatikou G, je definován jako $L(G) = \{w \in T^* | s \Rightarrow^* wf, \text{ kde } f \in F\}$.

Definice 5.2 *Levě rozšířená frontová gramatika* (viz též [9]) je šestice $G = (V, T, W, F, s, P)$, kde V, T, W, F a s mají stejný význam jako v případě běžné frontové gramatiky a $P \subseteq V \times (W - F) \times V^* \times W$ je konečná relace (poznamenejme, že tato definice nevyžaduje, aby pro každé $a \in V$ existovalo nějaké pravidlo $(a, b, x, c) \in P$). Kromě toho předpokládejme, že $\# \notin (V \cup W)$. Jestliže $u, v \in V^*\{\#\}V^*W$ jsou řetězce tvaru $u = w\#arb$, $v = wa\#rxc$, kde $a \in V$, $r, x, w \in V^*$, $b, c \in W$ a $(a, b, x, c) \in P$, potom píšeme

$$w\#arb \Rightarrow wa\#rxc[(a, b, x, c)]$$

Symboly $\Rightarrow^n$, $\Rightarrow^*$ a $\Rightarrow^+$ označují postupně derivaci délky n, $n \geq 0$, tranzitivní uzávěr a reflexivní-tranzitivní uzávěr relace přímé derivace $\vdash$.

Jazyk generovaný levě rozšířenou frontovou gramatikou G je definován jako $L(G) = \{y \in T^* | \#s \Rightarrow^* w\#yf$ pro nějaké $w \in V^*$ a $f \in F\}$.

Označme **QG** třídu jazyků generovaných frontovými gramatikami. Platí následující věta.

Věta 5.1 **QG** = **RE**. Důkaz je možné nalézt v [8].

Následující pomocná věta bude využita při konstrukci oboustranného zásobníkového automatu nad volnou grupou.

Lemma 5.2 Pro každý rekurzívně vyčíslitelný jazyk L existuje levě rozšířená frontová gramatika G taková, že $L = L(G)$ a pro každé pravidlo $(a, b, x, c) \in P$ platí $a \in (V - T)$, $b \in (W - F)$ a $x \in ((V - T)^* \cup T^*)$. Důkaz viz [3].

5.2 Oboustranné zásobníkové automaty nad volnými grupami

Definice 5.3 *Oboustranný zásobníkový automat nad volnou grupou* je n-tice $M = (Q, T, Z, R, z, Z_L, Z_R, F_M)$, kde

- Q je konečná množina stavů
- T je konečná vstupní abeceda
- Z je konečná zásobníková abeceda
- R je konečná množina pravidel tvaru $Z_1|Z_2px \to \gamma_1|\gamma_2q$, kde $Z_1, Z_2 \in Z$, $\gamma_1, \gamma_2 \in Z^\circ$, $x \in T^*$ a $p, q \in Q$; poznamenejme, že takovýto automat může načítat v jednom výpočetním kroku více než jeden vstupní symbol (transformace na klasický automat čtoucí nejvýše jeden symbol ze vstupu je triviální)
- z je počáteční stav
- Z_L resp. Z_R je počáteční symbol lévé resp. pravé strany zásobníku, $Z_L, Z_R \in Z$
- $F_M \subseteq Q$ je množina koncových stavů

Konfigurací oboustranného zásobníkového automatu nad volnou grupou rozumíme řetězec γqw, kde $q \in Q$, $\gamma \in Z^\circ$ a $w \in T^*$. Pokud $L\gamma Rpxy$ a $\gamma_L\gamma\gamma_Rqy$ jsou dvě konfigurace a $L|Rpx \to \gamma_L|\gamma_Rq \in R$, kde $x, y \in T^*$, $p, q \in Q$, $L, R \in Z$ a $\gamma_L, \gamma, \gamma_R \in Z^\circ$, pak automat M provádí *přechod* z konfigurace $L\gamma Rpxy$ do konfigurace $\gamma_L\gamma\gamma_Rqy$ podle pravidla $L|Rpx \to \gamma_L|\gamma_Rq$ a píšeme

$$L\gamma Rpxy \vdash \gamma_L\gamma\gamma_Rqy[L|Rpx \to \gamma_L|\gamma_Rq]$$

Relace $\vdash^n$, $\vdash^+$ a $\vdash^*$ označují posloupnost přechodů délky n, $n \geq 0$, tranzitivní a reflexivní-tranzitivní uzávěr relace $\vdash$ v tomto pořádí a jsou definovány obvyklým způsobem.

Jazyk přijímaný oboustranným zásobníkovým automatem M je definován jako $L(M) = \{w | w \in T^*, Z_R Z_L z w \vdash^* \varepsilon f, \text{ kde } f \in F_M\}$. Všimněme si, že řetězce vyskytující se na oboustranném zásobníku jsou tvořeny volnou grupou generovanou abecedou Z operací konkatenace. Řetězec w je automatem přijat pouze tehdy, pokud je beze zbytku načten, zásobník je prázdný a automat se po provedení posledního kroku nachází v některém koncovém stavu.

Nyní se již dostáváme k hlavnímu výsledku tohoto projektu. Označme $2\mathbf{PDA}^\circ$ třídu jazyků přijímaných oboustrannými zásobníkovými automaty nad volnými grupami.

Věta 5.3 $2\mathbf{PDA}^\circ = \mathbf{RE}$

Důkaz

Je dokázáno, že třída jazyků generovaných frontovými gramatikami (**QG**) je totožná s třídou rekurzívně vyčíslitelných jazyků (**RE**). Stačí tedy dokázat, že pro každou frontovou gramatiku $G = (V, T, W, F, Sq_0, P)$ je možné sestrojit oboustranný zásobníkový automat nad volnou grupou $M = (Q, T, Z, R, z, Z_L, Z_R, F_M)$ takový, že $L(G) = L(M)$. Bez ztráty obecnosti předpokládejme, že G splňuje podmínky popsané v Lemmatu 5.2.

Konstrukce Konstrukci oboustranného zásobníkového automatu nad volnou grupou provedeme aplikací následujících kroků:

- $Q = \{f, z\} \cup \{\langle q, 1\rangle, \langle q, 2\rangle | q \in W\}$
- $Z = \{Z_L, Z_R, \overline{Z_L}, \overline{Z_R}\} \cup (V - T) \cup \overline{N}$, kde $\overline{N} = \{\overline{x} | x \in (V - T)\}$
- $F_M = \{f\}$

Množina pravidel R je zkonstruována následujícím způsobem:

1) pro axiom Sq_0 gramatiky G, kde $S \in (V - T)$, $q_0 \in (W - F)$, přidej $Z_L | Z_R z \to Z_L | S Z_R \langle q_0, 1\rangle$ do R
2) pro každé $(A, q, x, p) \in P$, kde $A \in (V - T)$, $p, q \in (W - F)$, $x \in (V - T)^*$, přidej $Z_L | Z_R \langle q, 1\rangle \to Z_L \overline{A} | x Z_R \langle p, 1\rangle$ do R
3) pro každé $q \in W$ přidej $Z_L | Z_R \langle q, 1\rangle \to Z_L | Z_R \langle q, 2\rangle$ do R
4) pro každé $(A, q, y, p) \in P$, kde $A \in (V - T)$, $p, q \in (W - F)$, $y \in T^*$, přidej $Z_L | Z_R \langle q, 2\rangle y \to Z_L \overline{A} | Z_R \langle p, 2\rangle$ do R
5) pro každé $(A, q, y, t) \in P$, kde $A \in (V - T)$, $p \in (W - F)$, $t \in F$, přidej $Z_L | Z_R \langle q, 2\rangle y \to \overline{A} | \varepsilon f$ do R

Nyní je konstrukce hotova. Pro další části důkazu zaveďme následující notaci. Jestliže $\langle q, 1\rangle$ je aktuální stav automatu M, říkáme, že M je v módu *generování nonterminálů*.

Podobně jestliže $\langle q, 2\rangle$ je aktuální stav automatu M, říkáme, že M je v módu *čtení terminálů* (pro nějaké $q \in W$).

Nyní musíme dokázat rovnost $L(G) = L(M)$, tedy inkluze $L(G) \subseteq L(M)$ a $L(M) \subseteq L(G)$. Nejprve dokážeme první inkluzi, tedy $L(G) \subseteq L(M)$. Proveďme to postupnou demonstrací tvrzeních A, B a C.

Tvrzení A Jestliže $A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^i A_1 \dots A_n B_1 \dots B_i \# B_{i+1} \dots \dots B_m x_1 \dots x_i p$ v G, potom $Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle u, 1\rangle \omega \vdash^i Z_L \overline{B_i} \dots \dots \overline{B_1} \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m x_1 \dots x_i Z_R \langle p, 1\rangle \omega$ v M, kde $A_1, \dots, A_n$, $B_1, \dots, B_m \in (V - T)$, $x_1, \dots, x_i \in (V - T)^*$, $u, p \in (W - F)$, $m \geq 1$, $n \geq 0$, $\omega \in T^*$, $i < m$.

Základ indukce Nechť $i = 0$. Potom $A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^0 A_1 \dots \dots A_n \# B_1 \dots B_m u$ v G. Je zřejmé, že $Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle u, 1\rangle \omega \vdash^0 Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle u, 1\rangle \omega$ v M.

Indukční hypotéza Předpokládejme, že tvrzení A platí pro každé $i \leq l$, kde l je kladné celé číslo.

Indukční krok Uvažujme libovolnou derivaci tvaru $A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^{l+1} A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m x_1 \dots x_l x_{l+1} q$ a vyjádřeme ji přesněji jako $A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^l A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m x_1 \dots x_l p \Rightarrow A_1 \dots \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m x_1 \dots x_l x_{l+1} q$ v G, kde $l + 2 \leq m$.

Podle indukční hypotézy $Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle u, 1\rangle \omega \vdash^l Z_L \overline{B_l} \dots \dots \overline{B_1} \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m x_1 \dots x_l Z_R \langle p, 1\rangle \omega \vdash Z_L \overline{B_{l+1}} \overline{B_l} \dots \dots \overline{B_1} \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m x_1 \dots x_l x_{l+1} Z_R \langle q, 1\rangle \omega$ v M. V P existuje pouze jeden typ pravidel schopný provést derivaci $A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m x_1 \dots \dots x_l p \Rightarrow A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m x_1 \dots x_l x_{l+1} q$ v G. Jsou to pravidla tvaru $(B_{l+1}, p, x_{l+1}, q) \in P$, kde $B_{l+1} \in (V - T)$, $p, q \in (W - F)$ a $x_{l+1} \in (V - T)^*$. Podle druhého bodu konstrukce existuje v R pravidlo $Z_L | Z_R \langle p, 1\rangle \rightarrow Z_L \overline{B_{l+1}} | x_{l+1} Z_R \langle q, 1\rangle$, takže $Z_L \overline{B_l} \dots \overline{B_1} \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m x_1 \dots \dots x_l Z_R \langle p, 1\rangle \omega \vdash Z_L \overline{B_{l+1}} \overline{B_l} \dots \overline{B_1} \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m x_1 \dots \dots x_l x_{l+1} Z_R \langle q, 1\rangle \omega$ v M a tvrzení A tedy platí. □

Claim B Jestliže $A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \Rightarrow^i A_1 \dots A_n B_1 \dots \dots B_i \# B_{i+1} \dots B_m a_1 \dots a_k b_1 \dots b_i p$ v G, potom $Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots \dots B_m Z_R \langle u, 2\rangle b_1 \dots b_j \vdash^i Z_L \overline{B_i} \dots \overline{B_1} \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_i B_{i+1} \dots \dots B_m Z_R \langle p, 2\rangle b_{i+1} \dots b_j$ v M, kde $A_1, \dots, A_n, B_1, \dots, B_m \in (V - T)$, $a_1, \dots, a_k$, $b_1, \dots, b_j \in T^*$, $u, p \in (W - F)$, $i < m$, $i < j$, $k \geq 0$.

Základ indukce Nechť $i = 0$. Potom $A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \Rightarrow^0 A_1 \dots \dots A_n \# B_1 \dots B_m a_1 \dots a_k u$ v G. Je zřejmé, že také $Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots \dots B_m Z_R \langle u, 2\rangle b_1 \dots b_j \vdash^0 Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle u, 2\rangle b_1 \dots b_j$ v M.

Indukční hypotéza Předpokládejme, že tvrzení B platí pro všechna $i \leq l$, kde l je kladné celé číslo.

Indukční krok Uvažujme libovolnou derivaci tvaru $A_1 \dots A_n \# B_1 \dots B_m a_1 \dots \dots a_k u \Rightarrow^{l+1} A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m a_1 \dots a_k b_1 \dots b_l b_{l+1} q$ a vyjádřeme ji přesněji jako $A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \Rightarrow^l A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots \dots B_m a_1 \dots a_k b_1 \dots b_l p \Rightarrow A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m a_1 \dots a_k b_1 \dots \dots b_l b_{l+1} q$ v G, kde $l + 2 \leq m$, $k \geq 0$, $i < j$, $l + 2 \leq j$.

Podle indukční hypotézy $Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j\vdash^l$
$Z_L\overline{B_l}\dots\overline{B_1A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle p,2\rangle b_{l+1}\dots b_j\vdash$
$Z_L\overline{B_{l+1}B_l}\dots\overline{B_1A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle q,2\rangle b_{l+2}\dots b_j$ v M.

V tomto případě existuje pouze jedna možnost jak může G provést derivaci $A_1\dots A_nB_1\dots B_l\#B_{l+1}\dots B_ma_1\dots a_kb_1\dots b_lp\Rightarrow A_1\dots A_nB_1\dots$ $\dots B_lB_{l+1}\#B_{l+2}\dots B_ma_1\dots a_kb_1\dots b_lb_{l+1}q$, a to pomocí pravidla tvaru $(B_{l+1},p,b_{l+1},q)\in P$, kde $B_{l+1}\in(V-T)$, $p,q\in(W-F)$, $b_{l+1}\in T^*$. Podle čtvrtého bodu konstrukce existuje v R pravidlo $Z_L|Z_R\langle p,2\rangle b_{l+1}\to Z_L\overline{B_{l+1}}|Z_R\langle q,2\rangle$, takže $Z_L\overline{B_l}\dots\overline{B_1A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle p,2\rangle b_{l+1}\dots b_j\vdash Z_L\overline{B_{l+1}B_l}\dots$ $\dots\overline{B_1A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle q,2\rangle b_{l+2}\dots b_j$ v M a tvrzení B platí. $\square$

Claim C Jestliže $A_1\dots A_{n-1}\#A_nyq\Rightarrow A_1\dots A_{n-1}A_n\#yzt$ v G, kde $A_1,\dots$ $\dots,A_n\in(V-T)$, $y,z\in T^*$, $q\in(W-F)$, $t\in F$, potom $Z_L\overline{A_{n-1}}\dots\overline{A_1}A_1\dots$ $\dots A_nZ_R\langle q,2\rangle z\vdash\overline{A_n}\dots\overline{A_1}A_1\dots A_nf=\varepsilon f$ v M, kde $f\in F_M$.

Gramatika G provádí popsanou derivaci pomocí pravidla tvaru $(A_n,q,z,t)\in P$, kde $A_n\in(V-T)$, $z\in T^*$, $q\in(W-F)$, $t\in F$. Podle pátého bodu konstrukce existuje v R pravidlo $Z_L|Z_R\langle q,2\rangle z\to\overline{A_n}|\varepsilon f$, takže odpovídající krok popsaný tvrzením C se nepochybně v M vyskytuje. Tvrzení C tedy platí. $\square$

Dohromady tvrzení A, B a C dokazují, že skutečně $L(G)\subseteq L(M)$.

Abychom dokázali opačnou inkluzi, tedy $L(M)\subseteq L(G)$, demonstrujeme postupně platnost tvrzeních D, E a F.

Claim D Automat M přijímá každý řetězec $w\in L(M)$ následujícím způsobem.

$Z_LZ_Rzw_1w_2\dots w_r\vdash$
$Z_LSZ_R\langle q_0,1\rangle w_1w_2\dots w_r\vdash$
$Z_L\overline{S}SX_1^1X_2^1\dots X_{n_1}^1Z_R\langle q_1,1\rangle w_1w_2\dots w_r\vdash$
$Z_L\overline{X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2Z_R\langle q_2,1\rangle w_1w_2\dots w_r\vdash$
$Z_L\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3Z_R\langle q_3,1\rangle w_1w_2\dots w_r\vdash$

$\dots$
$\dots$
$\dots$

$Z_L\overline{X_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots$
$\dots X_1^mX_2^m\dots X_{n_m}^mZ_R\langle q_m,1\rangle w_1w_2\dots w_r\vdash$

$Z_L\overline{X_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots$
$\dots X_1^mX_2^m\dots X_{n_m}^mZ_R\langle q_m,2\rangle w_1w_2\dots w_r\vdash$

$Z_L\overline{X_{j+1}^kX_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots$
$\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots X_1^mX_2^m\dots X_{n_m}^mZ_R\langle q_{m+1},2\rangle w_2\dots w_r\vdash$

$Z_L\overline{X_{j+2}^kX_{j+1}^kX_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots$
$\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots X_1^mX_2^m\dots X_{n_m}^mZ_R\langle q_{m+2},2\rangle w_3\dots w_r\vdash$

$\dots$
$\dots$
$\dots$

$$Z_L\overline{X^m_{n_m-1}}\dots\dots\overline{X^k_{j+2}X^k_{j+1}X^k_j}\dots\overline{X^1_2X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots$$
$$\dots X^2_{n_2}X^3_1X^3_2\dots X^3_{n_3}\dots X^m_1X^m_2\dots X^m_{n_m}Z_R\langle q_{m+r-1},2\rangle w_r\vdash$$

$$\overline{X^m_{n_m}X^m_{n_m-1}}\dots\dots\overline{X^k_{j+2}X^k_{j+1}X^k_j}\dots\overline{X^1_2X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots$$
$$\dots X^2_{n_2}X^3_1X^3_2\dots X^3_{n_3}\dots X^m_1X^m_2\dots X^m_{n_m}f=\varepsilon f$$

v M, kde $w=w_1w_2\dots w_r$, $r\geq 1$, $w_1,\dots,w_r\in T^*$, $q_0,q_1,\dots,q_{m+r-1}\in(W-F)$, $X^1_1,\dots,X^1_{n_1},X^2_1,\dots,X^2_{n_2},\dots,X^m_1,\dots,X^m_{n_m}\in(V-T)$, $n_1,n_2,\dots,n_m\geq 0$, $m\geq 1$, $0\leq k\leq m$.

Proof of Claim D Nyní prozkoumáme všechny kroky konstrukce množiny automatových pravidel R. Poznamenejme, že v každém úspěšném výpočtu automat M používá vždy pravidla zkonstruovaná v kroku b před tím, než použije pravidla zkonstruovaná v kroku $b+1$, pro $b=1,\dots,4$.

V prvním kroku aplikuje M pravidlo $Z_L|Z_Rz\rightarrow Z_L|SZ_R\langle q_0,1\rangle$ zkonstruované v části 1, kde Sq_0 je axiom gramatiky G. Toto je jediný způsob, kterým může M provést přechod $Z_LZ_Rzw_1w_2\dots w_r\vdash Z_LSZ_R\langle q_0,1\rangle w_1w_2\dots w_r$. Všimněme si, že v každém úspěšném výpočtu automatu je toto pravidlo použito právě jednou. Pomocí něj se automat zároveň přepne do módu generování nonterminálů.

V další části výpočtu popsané sekvencí přechodů

$$Z_LSZ_R\langle q_0,1\rangle w_1w_2\dots w_r\vdash^*$$

$$Z_L\overline{X^k_j}\dots\overline{X^1_2X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots X^2_{n_2}X^3_1X^3_2\dots X^3_{n_3}\dots$$
$$\dots X^m_1X^m_2\dots X^m_{n_m}Z_R\langle q_m,1\rangle w_1w_2\dots w_r$$

používá M pravidla tvaru $Z_L|Z_R\langle q,1\rangle\rightarrow Z_L\overline{A}|xZ_R\langle p,1\rangle$ zkonstruovaná v bodu 2, kde $A\in(V-T)$, $x\in(V-T)^*$, $p,q\in(W-F)$. Tato část výpočtu je charakterizována stavy automatu tvaru $\langle q,1\rangle$, $q\in(W-F)$. Detailní důkaz této části je uveden v tvrzení E.

V kroku

$$Z_L\overline{X^k_j}\dots\overline{X^1_2X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots X^2_{n_2}X^3_1X^3_2\dots X^3_{n_3}\dots$$
$$\dots X^m_1X^m_2\dots X^m_{n_m}Z_R\langle q_m,1\rangle w_1w_2\dots w_r\vdash$$

$$Z_L\overline{X^k_j}\dots\overline{X^1_2X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots X^2_{n_2}X^3_1X^3_2\dots X^3_{n_3}\dots$$
$$\dots X^m_1X^m_2\dots X^m_{n_m}Z_R\langle q_m,2\rangle w_1w_2\dots w_r$$

automat M přepne pomocí pravidla tvaru $Z_L|Z_R\langle q,1\rangle\rightarrow Z_L|Z_R\langle q,2\rangle$ zkonstruovaného v bodu 3 do módu čtení nonterminálů. Poznamenejme, že toto pravidlo je během jednoho úspěšného výpočtu automatu použito právě jednou. Protože je jeho aplikací změněn aktuální stav automatu tvaru $\langle q,1\rangle$ na stav tvaru $\langle q,2\rangle$, $q\in(W-F)$, není již žádná další možnost použití pravidel zkonstruovaných v bodech 1 až 3.

V další části výpočtu popsané sekvencí kroků

$$Z_L\overline{X_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots$$
$$\dots X_1^mX_2^m\dots X_{n_m}^mZ_R\langle q_m,2\rangle w_1w_2\dots w_r\vdash^*$$

$$Z_L\overline{X_{n_m-1}^m}\dots\dots\overline{X_{j+2}^kX_{j+1}^kX_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots$$
$$\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots X_1^mX_2^m\dots X_{n_m}^mZ_R\langle q_{m+r-1},2\rangle w_r$$

používá M pravidla zkonstruovaná v bodu 4 a postupně načítá vstupní řetězec. Podrobný důkaz této části je uveden v tvrzení F.

Posledním krokem automat přejde do koncového stavu $f\in F_M$ pomocí nějakého pravidla tvaru $Z_L|Z_R\langle q,2\rangle y\to\overline{A}|\varepsilon f$ zkonstruovaného v bodu 5, kde $q\in(W-T)$, $y\in T^*$ a $A\in(V-T)$. Pokud je oboustranný zásobník po provedení tohoto kroku prázdný (po aplikaci grupových redukcí), pak byl vstupní řetězec přijat. V opačném případě řetězec přijat není, neboť v množině R neexistuje žádné pravidlo obsahující $f\in F_M$ na své levé straně. $\square$

Claim E Jestliže $Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,1\rangle\omega\vdash^i Z_L\overline{B_i}\dots$ $\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mx_1\dots x_iZ_R\langle p,1\rangle\omega$ v M, potom $A_1\dots A_n\#B_1\dots$ $\dots B_mu\Rightarrow^i A_1\dots A_nB_1\dots B_i\#B_{i+1}\dots B_mx_1\dots x_ip$ v G, kde $A_1,\dots,A_n,B_1,\dots$ $\dots,B_m\in(V-T)$, $x_1,\dots,x_i\in(V-T)^*$, $u,p\in(W-F)$, $i<m$.

Základ indukce Nechť $i=0$. Pak $Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,1\rangle\omega\vdash^0$ $Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,1\rangle\omega$ v M. Rozhodně také $A_1\dots A_n\#B_1\dots$ $\dots B_mu\Rightarrow^0 A_1\dots A_n\#B_1\dots B_mu$ v G.

Indukční hypotéza Předpokládejme, že tvrzení E platí pro všechna $i\le l$, kde l je kladné celé číslo.

Indukční krok Uvažujme libovolnou posloupnost kroků tvaru $Z_L\overline{A_n}\dots\overline{A_1}A_1\dots$ $\dots A_nB_1\dots B_mZ_R\langle u,1\rangle\omega\vdash^{l+1}Z_L\overline{B_{l+1}B_l}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mx_1\dots$ $\dots x_lx_{l+1}Z_R\langle q,1\rangle\omega$ a vyjádřeme ji přesněji jako $Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots$ $\dots B_mZ_R\langle u,1\rangle\omega\vdash^l Z_L\overline{B_l}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mx_1\dots x_lZ_R\langle p,1\rangle\omega\vdash$ $Z_L\overline{B_{l+1}B_l}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mx_1\dots x_lx_{l+1}Z_R\langle q,1\rangle\omega$ v M, kde $q\in$ $(W-F)$, $l+2\le m$.

Podle indukční hypotézy $A_1\dots A_n\#B_1\dots B_mu\Rightarrow^l A_1\dots A_nB_1\dots B_l\#B_{l+1}\dots$ $\dots B_mx_1\dots x_lp\Rightarrow A_1\dots A_nB_1\dots B_lB_{l+1}\#B_{l+2}\dots B_mx_1\dots x_lx_{l+1}q$ v G. V množině R je pouze jeden typ pravidel schopných provést posloupnost kroků $Z_L\overline{B_l}\dots$ $\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mx_1\dots x_lZ_R\langle p,1\rangle\omega\vdash Z_L\overline{B_{l+1}B_l}\dots\overline{B_1}\overline{A_n}\dots$ $\dots\overline{A_1}A_1\dots A_nB_1\dots B_mx_1\dots x_lx_{l+1}Z_R\langle q,1\rangle\omega$ v M. Jsou to pravidla tvaru $Z_L|Z_R\langle p,1\rangle\to Z_L\overline{B_{l+1}}|x_{l+1}Z_R\langle q,1\rangle\in R$. Podle konstrukce však existuje v G pravidlo tvaru $(B_{l+1},p,x_{l+1},q)\in P$, takže rovněž $A_1\dots A_n\#B_1\dots B_mu\Rightarrow^l$ $A_1\dots A_nB_1\dots B_l\#B_{l+1}\dots B_mx_1\dots x_lp\Rightarrow A_1\dots A_nB_1\dots B_lB_{l+1}\#B_{l+2}\dots$ $\dots B_mx_1\dots x_lx_{l+1}q$ v G a tvrzení E platí. $\square$

Claim F Jestliže $Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j\vdash^i Z_L\overline{B_i}\dots$ $\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_iB_{i+1}\dots B_mZ_R\langle p,2\rangle b_{i+1}\dots b_j$ v M, potom $A_1\dots$ $\dots A_n\#B_1\dots B_ma_1\dots a_ku\Rightarrow^i A_1\dots A_nB_1\dots B_i\#B_{i+1}\dots B_ma_1\dots a_kb_1\dots b_ip$ v G, kde $A_1,\dots,A_n,B_1,\dots,B_m\in V-T$, $a_1,\dots,a_k,b_1,\dots,b_j\in T^*$ a $C,D\in$ $W-F$, $i<m$.

Základ indukce Nechť $i = 0$. Pak $Z_L\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle u,2\rangle b_1\ldots b_j \vdash^0 Z_L\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle u,2\rangle b_1\ldots b_j$ v M. Je zřejmé, že $A_1\ldots A_n\#B_1\ldots \ldots B_ma_1\ldots a_ku \Rightarrow^0 A_1\ldots A_n\#B_1\ldots B_ma_1\ldots a_ku$ v G.

Indukční hypotéza Předpokládejme, že tvrzení F platí pro všechna $i \leq l$, kde l je kladné celé číslo.

Indukční krok Uvažujme libovolnou posloupnost kroků tvaru $Z_L\overline{A_n}\ldots\overline{A_1}A_1\ldots \ldots A_nB_1\ldots B_mZ_R\langle u,2\rangle b_1\ldots b_j \vdash^{l+1} Z_L\overline{B_{l+1}B_l}\ldots\overline{B_1A_n}\ldots\overline{A_1}A_1\ldots \ldots A_nB_1\ldots B_mZ_R\langle q,2\rangle b_{l+2}\ldots b_j$ a vyjádřeme ji přesněji jako $Z_L\overline{A_n}\ldots \ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle u,2\rangle b_1\ldots b_j \vdash^l Z_L\overline{B_l}\ldots\overline{B_1A_n}\ldots\overline{A_1}A_1\ldots \ldots A_nB_1\ldots B_mZ_R\langle p,2\rangle b_{l+1}\ldots b_j \vdash Z_L\overline{B_{l+1}B_l}\ldots\overline{B_1A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots \ldots B_mZ_R\langle q,2\rangle b_{l+2}\ldots b_j$ v M, kde $l+2 \leq j$.

Podle indukční hypotézy $A_1\ldots A_n\#B_1\ldots B_ma_1\ldots a_ku \Rightarrow^l A_1\ldots A_nB_1\ldots \ldots B_l\#B_{l+1}\ldots B_ma_1\ldots a_kb_1\ldots b_lp \Rightarrow A_1\ldots A_nB_1\ldots B_lB_{l+1}\#B_{l+2}\ldots \ldots B_ma_1\ldots a_kb_1\ldots b_lb_{l+1}q$ v G, kde $l+2 \leq m$.

V tomto případě existuje pouze jeden způsob jak může M provést krok $Z_L\overline{B_l}\ldots \ldots\overline{B_1A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle p,2\rangle b_{l+1}\ldots b_j \vdash Z_L\overline{B_{l+1}B_l}\ldots\overline{B_1A_n}\ldots \ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle q,2\rangle b_{l+2}\ldots b_j$ a sice pomocí nějakého pravidla tvaru $Z_L|Z_R\langle p,2\rangle b_{l+1} \to Z_L\overline{B_{l+1}}|Z_R\langle q,2\rangle \in R$. Podle konstrukce obsahuje P pravidlo tvaru (B_{l+1},p,b_{l+1},q), kde $B_{l+1} \in (V-T)$, $p,q \in (W-F)$, $b_{l+1} \in T^*$, takže $A_1\ldots A_nB_1\ldots B_l\#B_{l+1}\ldots B_ma_1\ldots a_kb_1\ldots b_lp \Rightarrow A_1\ldots A_nB_1\ldots B_lB_{l+1}\# B_{l+2}\ldots B_ma_1\ldots a_kb_1\ldots b_lb_{l+1}q$ v G a tvrzení F tedy platí. □

Tvrzení D, E a F dokazují, že skutečně $L(M) \subseteq L(G)$. Celkově potom $L(G) = L(M)$, čímž je věta dokázána. ■

5.3 Diskuse k zásobníkovým automatům

Podobně jako v případě konečných automatů, lze docílit zvýšení vyjadřovací schopnosti i pro zásobníkové automaty. Uvedená modifikace na dvouzásobníkový automat není jediná. Dokonce ani volná grupa, která generuje možné řetězce vyskytující se na zásobníku, není pro správnou funkci nutná. Místo ní bychom mohli přidat jednoduchá pravidla odstraňující vždy levý a pravý vrchol zásobníku. Tato pravidla by mohla být aplikována na samém konci výpočtu a zajišťovala by vyprázdnění zásobníku. Jakmile by nastala situace, že se levý a pravý vrchol zásobníku neshodují, celý proces by byl zablokován a výsledkem by bylo nepřijetí vstupního řetězce. Naopak, v případě úplného vyprázdnění zásobníku a přechodem do koncového stavu, by byl řetězec přijat.

Dalším vylepšením tohoto automatu může být například redukce počtu symbolů zásobníkové abecedy. Každý symbol by byl poté kódován vhodným binárním kódem.

Uvedený přístup pro zvýšení vyjadřovacích schopností zásobníkových automatů však není jediný. Další modifikací se zabývá [9]. Zde autoři představují tzv. řízené zásobníkové automaty. S využitím vhodného řídícího jazyka opět zvyšují akceptační schopnosti běžných zásobníkových automatů. Případné zájemce o tuto problematiku odkazujeme na [9].

6 Závěr

Prezentovaný dokument ukazuje, že současné automaty mohou tvořit jakýsi základ pro další formální modely a vhodnými modifikacemi, omezeními a rozšířeními můžeme zvyšovat jejich vyjadřovací schopnosti. Poslední prezentovaný výsledek navíc ukazuje, že vhodnou modifikací běžného zásobníkového automatu dosáhneme u nově vzniklého modelu síly Turingova stroje. Případné modifikace Turingova stroje za účelem možného zvýšení jeho vyjadřovacích schopností jsou však již bezpředmětné, neboť jak plyne z Turingovy teze, za tuto hranici se již nelze dostat. Příkladem může být vícepáskový Turingův stroj. Přestože ten určitým způsobem rozšiřuje paměťové možnosti běžného jednopáskového Turingova stroje, lze každý vícepáskový Turingův stroj transformovat na ekvivalentní Turingův stroj jednopáskový. Vyjadřovací schopnost tedy nebyla žádným způsobem ovlivněna.

Obdobný přístup ke zvyšování vyjadřovacích schopností je však aplikovatelný nejen v oblasti automatů, ale taktéž v oblasti gramatik. Stačí jen zvolit vhodný základ. Vezmeme-li například bezkontextovou gramatiku, můžeme zavedením určitého stupně řízení taktéž zvýšit její sílu. Příkladem tohoto mohou být například programované a maticové gramatiky, gramatické systémy, nebo bezkontextové gramatiky nad volnými grupami prezentované v jiném projektu.

Reference

[1] Salomaa, A.: *Formal Languages.* Academic Press, New York, 1973.

[2] Jacobson, N.: *Basic Algebra, 2nd ed.* W.H. Freeman, New York, 1989.

[3] Meduna, A.: *Automata and Languages: Theory and Applications.* Springer, London, 2000.

[4] The Accepting Power of Finite Automata Over Groups. *TUCS Technical Report No 70*, November 1996.

[5] Dassow, J., Mitrana, V.: Finite Automata Over Free Groups. *Int. Journal of Algebra and Computation Vol. 10*, No. 6, 2000, pp. 725-737.

[6] Meduna, A.: Simultaneously One-Turn Two-Pushdown Automata. *International Journal of Computer Mathematics, No. 82* Taylor & Francis Informa plc, GB, 2003, pp. 1-9.

[7] Păun, G.: *A New Generative Device: Valence Grammars.* Rev Roum. Math. Pures et Appl. **25** (6) (1980), pp. 911-924. *Acta Cybernetica, 14*, 2000, pp. 653-664.

[8] Kleijn, H. C. M., Rozenberg, G.: On the Generative Power of Regular Pattern Grammars. *Acta Informatica*, 20 (1983), pp. 391-411.

[9] Meduna, A., Kolář, D.: Regulated Pushdown Automata. *Acta Cybernetica*, 14 (2000), pp. 653-664.