VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

FORMÁLNÍ MODELY NAD VOLNÝMI GRUPAMI
FORMAL MODELS OVER FREE GROUPS

TEZE DISERTAČNÍ PRÁCE
SHORT PHD THESIS

AUTOR PRÁCE Ing. PETR BLATNÝ
AUTHOR

VEDOUCÍ PRÁCE prof. RNDr. ALEXANDER MEDUNA, CSc.
SUPERVISOR

BRNO 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

FORMÁLNÍ MODELY NAD VOLNÝMI GRUPAMI

FORMAL MODELS OVER FREE GROUPS

TEZE DISERTAČNÍ PRÁCE
SHORT PHD THESIS

AUTOR PRÁCE — Ing. PETR BLATNÝ
AUTHOR

VEDOUCÍ PRÁCE — prof. RNDr. ALEXANDER MEDUNA, CSc.
SUPERVISOR

BRNO 2007

Abstrakt

Práce pojednává o nových formálních modelech popisující rekurzivně vyčíslitelné jazyky. Jsou představeny konstrukce bezkontextových a E0L gramatik nad volnou grupou a oboustranný zásobníkový automat nad volnou grupou. Dále bude následovat také pojednání o jejich redukovaných verzích. V případě bezkontextových gramatik nad volnou grupou jsme schopni redukovat počet nonterminálních symbolů na osm, aniž bychom snížili sílu těchto gramatik. Lepšího výsledku dosáhneme v případě E0L gramatik nad volnou grupou, kdy nám stačí k popisu rekurzivně vyčíslitelného jazyka E0L gramatika nad volnou grupou s šesti nonterminálními symboly. V oblasti automatů se podařilo redukovat zásobníkovou abecedu na pouhé čtyři nonterminální symboly.

Klíčová slova

bezkontextová gramatika, E0L gramatika, derivace, volná grupa, oboustranný zásobníkový automat, bezkontextová gramatika nad volnou grupou, E0L gramatika nad volnou grupou, oboustranný zásobníkový automat nad volnou grupou, Kurodova normální forma, frontová gramatika, zleva rozšířená frontová gramatika

Abstract

In the context-free and E0L grammars discussed in this paper, the derivations are introduced over free groups rather than free monoids. It is proved that both grammars with derivations introduced in this way characterize the family of recursively enumerable languages in a very succinct way. Specifically, this characterization is based on the eight-nonterminal context-free grammars and six-nonterminal E0L grammars over free groups. The next part of the paper establishes the two-sided pushdown automata over free groups. Their construction is based on queue grammars. A version with the reduced number of four symbols in the pushdown alphabet is also studied.

Keywords

context-free grammar, E0L grammar, derivation, free group, two-sided pushdown automata, context-free grammar over free group, E0L grammar over free group, two-sided pushdown automata over free group, Kuroda normal form, queue grammar, left-extended queue grammar

Citace

Petr Blatný: Formální modely nad volnými grupami, teze disertační práce, Brno, FIT VUT v Brně, 2007

Formální modely nad volnými grupami

Prohlášení

Prohlašuji, že jsem disertační práci vypracoval samostatně pod vedením školitele, prof. RNDr. Alexandera Meduny, CSc. Většina uvedených výsledků byla dosažena společně s kolegou Ing. Radkem Bidlem a s mým školitelem. Dále jsou zde obsaženy i některé výsledky od jiných autorů. Vždy jsem však uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

....
Petr Blatný
30. května 2007

Poděkování

Na tomto místě bych rád poděkoval svému školiteli prof. RNDr. Alexandru Medunovi, CSc za odborné vedení, inspiraci a ochotu vždy dobře poradit. Poděkování také patří Ing. Radku Bidlovi, který se podílel na většině článků týkajících se formálních modelů nad volnými grupami a stal se tak neocenitelným kolegou v tomto výzkumu.

Curriculum Vitae

Jméno:	Petr Blatný
Narozen:	30.11.1980
Vzdělání:	2004—Ing. v oboru výpočetní technika a informatika Fakulta informačních technologií Vysoké učení technické v Brně nyní—Fakulta informačních technologií Vysoké učení technické v Brně (Doktorský studijní program)
Výuka:	příprava písemných zkoušek v předmětu Formální jazyky a překladače příprava písemných zkoušek v předmětu Výstavba překladačů vedení bakalářských prací a diplomové práce
Publikace:	Autor/spoluautor příspěvků na konferencích Student EEICT 2003, 2004, 2005, 2006 Autor příspěvku na mezinárodní konferenci ISIM 2005 Spoluautor příspěvku na mezinárodní konferenci WFM 2007 Spoluautor příspěvku na konferencích MEMICS 2005, MEMICS 2006
Časopisecké publikace:	Spoluautor příspěvků v mezinárodních časopisech Kybernetika, Vol. *2007*, No. 1, 2007 Schedae Informaticae, Krakov, 2007

Obsah

Kapitola 1

Úvod

V teorii formálních jazyků se představuje velké množství modifikací základních formálních modelů. Tyto modifikace mají za cíl zvýšit vyjadřovací schopnosti daného modelu a stejně tak je to i v této práci. Zde představené modifikace zasahují do definice derivace, derivace je definována nad volnou grupou místo nad volným monoidem a žádná jiná režie nebo řízení generování vět jazyka není nutná. Využívá se tedy především vlastností volných grup, konkrétně se jedná o využití konkatenace vzájemně inverzních prvků.

1.1 Klasifikace gramatik a jazyků

V současné době je známo velké množství různých struktur pro popis formálních jazyků. Nejčastěji se jedná o popis pomocí gramatiky. Na základě tvaru přepisovacích pravidel těchto gramatik, lze jazyky generované gramatikou rozdělit do čtyř tříd definované Chomského hierarchií jazyků. Existují ovšem různé modifikace gramatik, které nelze zařadit přímo do jedné z definovaných tříd. Takovéto gramatiky mohou zasahovat do několika tříd současně a přitom žádnou z nich nepokrývají. Přesto je Chomského hierarchie nejpoužívanějším měřítkem generativní síly gramatik a jejich jazyků.

Na samém vrcholu Chomského hierarchie jsou tzv. jazyky typu 0. V některé literatuře se můžeme též setkat s označením rekurzivně vyčíslitelné jazyky, frázově strukturované jazyky nebo též jazyky přijímané Turingovými stroji. Všechny výše uvedená označení jsou si ekvivalentní. Tato množina jazyků je v této práci označována zkratkou **RE**. Rekurzivně vyčíslitelné jazyky jsou nejčastěji popisovány Turingovými stroji nebo gramatikami typu 0, které se někdy označují také jako obecné nebo neomezené gramatiky. Tyto gramatiky povolují přepisovací pravidla, která mají na své levé straně více než jeden symbol, o takovýchto pravidlech se v této práci zmiňujeme, že jsou *kontextová*. Například přepisovací pravidlo $AB \to CD$ je pravidlo kontextového tvaru.

Nejznámější třídou jazyků v praktické oblasti jsou jazyky bezkontextové, označované také jako typ 2. Tato třída jazyků je v této práci označována zkratkou **CF**. Nejčastější formální modely pro popis bezkontextových jazyků jsou bezkontextové gramatiky nebo zásobníkové automaty. Jak už jejich název napovídá, nedovolují tyto gramatiky přepisovací pravidla v kontextovém tvaru. Dostáváme se tedy k dalšímu pojmu a to jsou bezkontextová pravidla. Každé pravidlo, které má na své levé straně pouze jediný symbol, nazýváme v této práci jako *bezkontextové*. Například přepisovací pravidlo $A \to x$ je bezkontextového tvaru. Bezkontextové jazyky jsou základem pro většinu programovacích jazyků a jsou podle nich navrženy syntaktické analyzátory pro překladače k daným jazykům.

1.2 Předmět výzkumu disertační práce

Standardně jsou derivace v gramatikách definovány nad volným monoidem generovaným množinami terminálních a nonterminálních symbolů (totální abecedou) operací konkatenace. Avšak v současné době jsou v některých studií tyto derivace definovány nad jinými algebraickými strukturami (viz. [20], [21], [18]). V této práci provedeme modifikaci derivace a to tak, že ji budeme definovat nad volnou grupou. Dále se budeme zabývat generativní silou takto modifikovaných formálních modelů a možné redukce počtu nonterminálních symbolů a zásobníkové abecedy v případě automatů.

Současný stav poznání v této oblasti, rozumí se formální modely pouze nad volnými grupami, nepřináší kromě této práce zásadní výsledky spadající do teoretické informatiky. Výjimkou jsou práce zabývající se konečnými automaty nad volnou grupou, které jsou schopny charakterizovat třídu bezkontextových jazyků. Definice těchto konečných automatů nad volnou grupou a jejich generativní schopnosti lze nalézt v [24] a [11]. Z oblasti automatů je také diplomová práce Michala Berky [1], ve které jsem byl školitelem. Tato práce byla zaměřena na návrh a implementaci syntaktického analyzátoru založeného na formálním modelu nad volnou grupou.

Tento dokument pojednává o návrhu struktur, které jsou schopny generovat celou třídu rekurzivně vyčíslitelných jazyků. Jako základ použijeme bezkontextovou gramatiku, E0L gramatiku a v posledním případě rozšířený oboustranný zásobníkový automat. Společně definujeme volnou grupu nad totální abecedou těchto gramatik a nad zásobníkovou abecedou oboustranného zásobníkového automatu. Využitím schopností uvedených prostředků lze dokázat, že bezkontextové gramatiky nad volnými grupami, E0L gramatiky nad volnými grupami a rozšířené oboustranné zásobníkové automaty nad volnými grupami definují třídu rekurzivně vyčíslitelných jazyků a zachovávají si tvar přepisovacích pravidel z původních modelů. Popis a vlastnosti těchto modelů byl prezentován v [4].

Protože zde představené základní modifikace zvyšují počet nonterminálních symbolů, týká se další výzkum redukce počtu nonterminálních symbolů, to v případě bezkontextových a E0L gramatik nad volnou grupou. V případě rozšířených oboustranných zásobníkových automatů redukujeme počet symbolů zásobníkové abecedy. Je dokázáno, že všechny modifikace založené na redukci počtu nonterminálních symbolů zachovávají svoji vyjadřovací sílu, tedy že stále popisují rekurzivně vyčíslitelnou třídu jazyků. Následuje přehled dosažených výsledků v případě redukce počtu nonterminálních symbolů:

- Počet nonterminálních symbolů bezkotextové gramatiky nad volnou grupou byl redukován na osm symbolů.
- Počet nonterminálních symbolů E0L gramatiky nad volnou grupou byl redukován na šest symbolů.
- Počet nonterminálních symbolů zásobníkové abecedy rozšířeného oboustranného zásobníkového automatu byl redukován na čtyři symboly.

Kapitola 2

Modely pro popis jazyků

V této kapitole budou představeny formální modely potřebné pro budoucí konstrukce gramatik a zásobníků nad volnou grupou.

2.1 L systémy

L systémy poprvé představil Aristid Lindenmayer v roce 1968 [17]. Původně se jednalo o paralelní přepisovací systém modelující vícebuněčné organismy. Základní myšlenka těchto systémů našla využití i v teoretické informatice. Vznikají nové jazyky založené na 0L, D0L, P0L, E0L a dalších gramatikách. Zvláštností většiny tříd jazyků, které jsou přijímány gramatikami založenými na 0L systémech, je jejich pozice v Chomského hierarchii - jednotlivé třídy mohou překrývat a přitom žádnou z nich nepokrývají.

Nás bude zajímat třída jazyků přijímaná E0L gramatikami, která leží nad třídou bezkontextových jazyků, ale nepokrývá celou třídu jazyků kontextových. Vhodnou modifikací, která bude představena později, se pokusíme sílu jazyků E0L gramatik zvýšit na sílu turingových strojů.

Poznámka: další gramatiky založené na 0L systémech (P0L, D0L, ET0L, CT0L, FEP0L, atd.) a vztahy mezi nimi lze najít v [25] a [23].

E0L systém

E0L systém, nebo také E0L gramatika, je čtveřice $G = (V, \Sigma, P, w)$, kde

- V je konečná množina symbolů
- $\Sigma \subseteq V$ je konečná podmnožina terminálních symbolů
- P je konečná množina pravidel tvaru $a \to x, \quad a \in V, \quad x \in V^*$
- $w \in V^*$ je počáteční (startovací) řetězec, označuje se také jako *axiom* gramatiky

Definice 2.1.1 Nechť $G = (V, \Sigma, P, w)$ je E0L systém a nechť λ a μ jsou řetězce z V^*. Mezi řetězci λ a μ platí relace $\Rightarrow_G$, nazývaná *přímá derivace*, jestliže můžeme řetězce λ a μ vyjádřit ve tvaru

$$\lambda = x_1 x_2 \dots x_n$$

$$\mu = y_1 y_2 \dots y_n$$

$$\text{kde } x_i \in V, \quad y_i \in V^* \text{ a } x_i \to y_i \in P \text{ pro } 1 \le i \le n.$$

Platí-li mezi řetězci λ a μ relace přímé derivace, pak píšeme $\lambda \Rightarrow_G \mu$ a říkáme, že řetězec μ lze přímo generovat z řetězce λ v systému G.

Definice 2.1.2 Nechť $G = (N, \Sigma, P, w)$ je E0L systém a λ a μ jsou řetězce z V^*. Mezi řetězci λ a μ platí relace $\Rightarrow^+$ nazývaná *derivace*, jestliže existuje posloupnost přímých derivací $v_{i-1} \Rightarrow v_i \quad 1 \le i \le n$ taková, že platí:

$$\lambda = v_0 \Rightarrow v_1 \Rightarrow \dots \Rightarrow v_{n-1} \Rightarrow v_n = \mu.$$

Tuto posloupnost nazýváme *derivací délky n*. Platí-li $\lambda \Rightarrow^+ \mu$, pak říkáme, že řetězec μ lze *generovat* z řetězce λ v systému G. Relace $\Rightarrow^+$ je tranzitivním uzávěrem relace přímé derivace $\Rightarrow$. Symbolem $\Rightarrow^n$ značíme n-tou mocninu relace $\Rightarrow$.

Definice 2.1.3 Jestliže v systému G platí pro řetězce λ a μ relace $\lambda \Rightarrow^+ \mu$ nebo identita $\lambda = \mu$, pak píšeme $\lambda \Rightarrow^* \mu$. Relace $\Rightarrow^*$ je tranzitivním a reflexivním uzávěrem relace přímé derivace $\Rightarrow$.

Definice 2.1.4 Nechť $G = (V, \Sigma, P, w)$ je E0L systém. Řetězec $\alpha \in V^*$ nazýváme *větnou formou*, jestliže platí $S \Rightarrow^* \alpha$, tj. řetězec α je generovatelný z axiomu w. Větná forma, která obsahuje pouze terminální symboly, se nazývá *věta*. *Jazyk* $L(G)$, generovaný systémem G, je definován množinou všech vět

$$L(G) = \{u | w \Rightarrow^* u, \quad u \in \Sigma^*\}.$$

2.2 Frontové gramatiky

Na konci této práce budou představeny oboustranné zásobníkové automaty nad volnou grupou, k jejich konstrukci bude zapotřebí definovat následující pojmy: zleva rozšířená frontová gramatika a rozšířený zásobníkový automat, o těchto formálních modelech pojednává tato a následující podkapitola.

Frontová gramatika

Definice 2.2.5 *Frontová gramatika* (viz [14]) je šestice, $Q = (V, T, W, F, s, P)$, kde V a W jsou totální abecedy takové, že $V \cap W = \emptyset$, $T \subseteq V$, $F \subseteq W$, $s \in (V - T)(W - F)$ je výchozí (startovací) axiom a $P \subseteq V \times (W - F) \times V^* \times W$ je konečná relace taková, že pro každé $a \in V$ existuje nějaký prvek $(a, b, x, c) \in P$.

Jestliže $u, v \in V^*W$, $u = arb$, $v = rxc$, $a \in V$, $r, x \in V^*$, $b, c \in W$ a $(a, b, x, c) \in P$, potom $u \Rightarrow v$ v Q. Symbolem $\Rightarrow$ je označována *relace přímé derivace* v Q. Obvyklým způsobem jsou pak definovány relace $\Rightarrow^n$, $\Rightarrow^+$ a $\Rightarrow^*$.

Jazyk generovaný frontovou gramatikou Q, $L(Q)$, je definován jako

$$L(Q) = \{w | s \Rightarrow^* wf, w \in T^*, f \in F\}.$$

Konstrukce oboustranného zásobníkového automatu nad volnou grupou vychází z modifikace této frontové gramatiky, která byla představena v [22].

Označme třídu jazyků generovaných frontovými gramatikami pomocí **QG**. Potom platí následující věta.

Věta 2.2.1 **QG** = **RE**. Důkaz je možné nalézt v [14].

Zleva rozšířená frontová gramatika

Definice 2.2.6 *Zleva rozšířená frontová gramatika* je šestice, $Q = (V, T, W, F, s, P)$, kde V, T, W, F a s mají stejný význam jako v případě frontové gramatiky. Množina pravidel P je definována jako konečná relace $P \subseteq V \times (W - F) \times V^* \times W$. Na rozdíl od frontové gramatiky striktně nevyžaduje, aby pro každé $a \in V$ existoval nějaký prvek (a, b, x, c) v P.

Předpokládejme dále, že $\# \notin V \cup W$. Jestliže $u, v \in V^*\{\#\}V^*W$, kde $u = w\#arb$, $v = wa\#rxc$, $a \in V$, $r, x, w \in V^*$, $b, c \in W$ a $(a, b, x, c) \in P$, potom $u \Rightarrow v$ v Q. Standardním způsobem jsou pak také definovány relace $\Rightarrow^n$, kde $n \geq 0$, $\Rightarrow^+$ a $\Rightarrow^*$.

Jazyk generovaný zleva rozšířenou frontovou gramatikou je definován jako

$$L(Q) = \{v | \#s \Rightarrow^* w\#vf, w \in V^*, v \in T^*, f \in F\}.$$

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

Lemma 2.1 Pro každý rekurzivně vyčíslitelný jazyk L existuje zleva rozšířená frontová gramatika Q taková, že $L = L(Q)$ a pro každé pravidlo $(a, b, x, c) \in P$ platí $a \in (V - T)$, $b \in (W - F)$, $x \in ((V - T)^* \cup T^*)$. Důkaz viz [19].

2.3 Zásobníkové automaty

Poslední formální model, který v této práci bude modifikován je zásobníkový automat, v tomto případě kromě přidání vlastností volné grupy se bude modifikace týkat i samotné struktury modelu.

Bude změněn zásobník a to tak, aby byl schopný vkládat symboly nebo řetězce vstupní abecedy z obou stran. Vznikne tak modifikace nazvaná oboustranný zásobníkový automat a rozšířený oboustranný zásobníkový automat. Pro účel konstrukce obou zmíněných modifikací uvádím pro úplnost definice nemodifikovaných verzí zásobníkových automatů.

Zásobníkový automat

Definice 2.3.7 *Zásobníkový automat* je sedmice $M = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

- Q je konečná množina stavů
- Σ je konečná vstupní abeceda
- Γ je konečná zásobníková abeceda

- R je konečná množina pravidel tvaru $Apa \rightarrow wq$, kde $A \in \Gamma$, $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ a $w \in \Gamma^*$
- $s \in Q$ je počáteční stav
- $S \in \Gamma$ je počáteční symbol zásobníku
- $F \subseteq Q$ je množina koncových stavů

Konfigurací zásobníkového automatu M rozumíme řetězec xpy, kde $p, q \in Q$, $A \in \Gamma$, $x, v, w \in \Gamma^*$, $a \in \Sigma$ a $y \in \Sigma^*$. Pokud $xApay$ a $xwqy$ jsou dvě konfigurace a $Apa \rightarrow wq \in R$, pak automat M provádí *přechod* z konfigurace $xApay$ do konfigurace $xwqy$ podle pravidla $Apa \rightarrow wq$ a píšeme

$$xApay \vdash_M xwqy[Apa \rightarrow wq]$$

nebo stručněji $xApay \vdash xwqy$. 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:

1. přechodem do koncového stavu:
   $L(M_f) = \{w | w \in \Sigma^* \wedge Ssw \vdash^* rf$, kde $f \in F, r \in \Gamma^*\}$
2. s vyprázdněním zásobníku:
   $L(M_e) = \{w | w \in \Sigma^* \wedge Ssw \vdash^* p$, kde $p \in Q\}$
3. přechodem do koncového stavu a s vyprázdněním zásobníku:
   $L(M_{fe}) = \{w | w \in \Sigma^* \wedge Ssw \vdash^* f$, kde $f \in F\}$

Rozšířený zásobníkový automat

Definice 2.3.8 *Rozšířený zásobníkový automat* je sedmice $N = (Q, \Sigma, \Gamma, R, s, S, F)$, kde

- Q je konečná množina stavů
- Σ je konečná vstupní abeceda
- Γ je konečná zásobníková abeceda
- R je konečná množina pravidel tvaru $vpa \rightarrow wq$, kde $p, q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, a $v, w \in \Gamma^*$
- $s \in Q$ je počáteční stav
- $S \in \Gamma$ je počáteční symbol zásobníku
- $F \subseteq Q$ je množina koncových stavů

Konfigurací zásobníkového automatu N rozumíme řetězec xpy, kde $p, q \in Q$, $x, v, w \in \Gamma^*$, $a \in \Sigma$ a $y \in \Sigma^*$. Pokud $xvpay$ a $xwqy$ jsou dvě konfigurace a $vpa \rightarrow wq \in R$, pak automat N provádí *přechod* z konfigurace $xvpay$ do konfigurace $xwqy$ podle pravidla $vpa \rightarrow wq$ a píšeme

$$xvpay \vdash_N xwqy[vpa \rightarrow wq]$$

nebo stručněji $xvpay \vdash xwqy$. 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 N je definován třemi způsoby stejně jako zásobníkový automat M. Rozšíření tohoto zásobníkového automatu spočívá ve schopnosti číst ze vstupu nejen symboly ale i řetězce.

Kapitola 3

Volné grupy

Definujme si konečně strukturu volné grupy, která nás bude v souvislosti s naším výzkumem zajímat. Uveďme nejprve, jak je definována volná varianta struktury jednodušší — tedy monoidu.

3.1 Volný monoid

Volný monoid na množině V je monoid, jehož prvky jsou všechny konečné řetězce složené z prvků množiny V za pomoci binární operace konkatenace včetně prázdného řetězce ε, který je neutrálním prvkem. Tyto řetězce často nazýváme *slova* a množinu všech slov nad množinou V značíme V^*.

Definujme nyní výše uvedené formálně.

Definice 3.1.1 Množina V se nazývá *volná báze* monoidu G, jestliže V generuje G a každé zobrazení $f : V \to H$, kde H je monoid, lze rozšířit na homomorfismus $g : G \to H$. Monoid G se nazývá *volný*, jestliže má alespoň jednu volnou bázi.

Volný monoid generovaný množinou generátorů V pomocí operace konkatenace je v teorii formálních jazyků značen jednoduše symbolem V^*.

Příklad 3.1.1 Uvažujme množinu $V = \{a, b, c\}$. Potom prvky volného monoidu budou všechny řetězce složené z prvků množiny V. Například abc, ε, $abcc$, a, $cbcc \in V^*$.

3.2 Volná grupa

Podobně jako v případě volného monoidu, definujme si pojem volné grupy nejprve neformálně. *Volná grupa* na množině M je grupa, jejíž prvky jsou všechny konečné řetězce složené z prvků množiny M za pomoci binární operace konkatenace a to včetně prázdného řetězce ε, který je neutrálním prvkem. Připomeňme, že aby se jednalo o grupu, musí množina M ke každému svému prvku obsahovat prvek inverzní (výjimku tvoří pouze neutrální prvek, který je inverzní sám k sobě).

Uveďme si formální definici.

Definice 3.2.2 Množina V se nazývá *volná báze* grupy G, jestliže V generuje G a každé zobrazení $f : V \to H$, kde H je grupa, lze rozšířit na homomorfismus $g : G \to H$. Grupa G se nazývá *volná*, jestliže má (alespoň jednu) volnou bázi.

Aby nedošlo k záměně označení volného monoidu a volné grupy, stanovme si zde, že volnou grupu generovanou množinou generátorů V pomocí operace konkatenace budeme značit jednoduše symbolem V°. Protože množina V obsahuje ke každému prvku i prvek inverzní, mohou se všechny prvky z V vyskytovat v jednotlivých slovech množiny V°. Každé slovo, které obsahuje dvojice $\overline{x}x$ nebo $x\overline{x}$, lze dále redukovat až k jejich úplnému odstranění. Slovo, které neobsahuje žádné výše uvedené dvojice prvků vzájemně inverzních, se potom nazývá *redukované*. Lze dokázat, že každému slovu odpovídá pouze jediné slovo redukované a to bez ohledu na pořadí redukcí jednotlivých dvojic vzájemně inverzních symbolů. Důkaz je možné najít např. v [12].

V pozdějším spojení gramatiky a volné grupy však budeme možnost redukce dvojic vzájemně inverzních symbolů hojně využívat. Zvláště důležitým prvkem se stane inverzní řetězec.

Definice 3.2.3 Nechť $w = w_1 \ldots w_n$ je řetězec z V°, kde $w_1, \ldots w_n \in V$, $n \geq 0$, potom *inverzní řetězec* k řetězci w je definován jako $\overline{w} = \overline{w_n} \ldots \overline{w_1}$, kde $\overline{w_1}, \ldots, \overline{w_n} \in V$ a tedy $\overline{w} \in V^\circ$.

Příklad 3.2.2 Uvažujme množinu $V = \{a, b, c, \overline{a}, \overline{b}, \overline{c}\}$, potom inverzní řetězec k řetězci $bcaa \in V^\circ$ je roven řetězci $\overline{aacb} \in V^\circ$. Protože $\overline{a}a = a\overline{a} = \varepsilon$, $\overline{b}b = b\overline{b} = \varepsilon$ a $\overline{c}c = c\overline{c} = \varepsilon$, je zřejmé, že spojením obou řetězců získáme $bcaa\overline{aacb} = \overline{aacb}bcaa = \varepsilon$.

Definice 3.2.4 Nechť w je řetězec z V° a $\overline{w}$ je inverzní řetězec k řetězci w. Při jejich konkatenaci dojde k následnému vyrušení $\overline{w}w = \varepsilon$ a $w\overline{w} = \varepsilon$, tento jev založený na konkatenaci vzájemně inverzních symbolů budeme označovat jako *redukci*.

Příklad 3.2.3 Uvažujme opět množinu $V = \{a, b, c, \overline{a}, \overline{b}, \overline{c}\}$, kde inverzní prvek ke každému prvku $x \in V$ je prvek $\overline{x} \in V$. Každý prvek množiny V má tedy svůj inverzní protějšek a tak lze množinu V použít jako generátor volné grupy V°. Prvky volné grupy V° jsou všechny řetězce složené z prvků množiny V. Například abc, ε, $\overline{a}\overline{b}\overline{c}$, $a\overline{b}\overline{c}$, $c\overline{a}\overline{c}bcc \in V^\circ$.

Kapitola 4

Gramatiky nad volnými grupami

V této kapitole budou představeny nové formální modely založené na gramatikách. Standardně jsou derivace v gramatikách definovány nad volným monoidem generovaným množi-nami terminálních a nonterminálnich symbolů (totální abecedou) operací konkatenace. V této práci provedeme modifikaci derivace a to tak, že ji budeme definovat nad volnou grupou. Dále se budeme zabývat generativní silou takto modifikovaných gramatik. Jako první případ prozkoumáme modifikaci bezkontextových gramatik. Budeme vycházet z gramatik typu 0, popsaných Kurodovou normální formou, představíme konstrukci, která danou gramatiku v Kurodově normální formě transformuje na gramatiku bezkontextovou nad volnou grupou. Podobně budeme transformovat i E0L gramatiky.

Tato kapitola byla ve velkém měřítku inspirovaná [20].

4.1 Bezkontextové gramatiky nad volnými grupami

Výše jsme si uvedli matematické a algebraické základy formálně, zde zavedeme určité zjednodušení. Uvažujme libovolnou abecedu V. Zdůrazněme znovu, že symbolem V° budeme označovat volnou grupu generovanou množinou generátorů V operací konkatenace. Přitom platí, že pro každý symbol $x \in V$ existuje právě jeden inverzní symbol $\overline{x} \in V$. Prázdný řetězec $\varepsilon \in V^\circ$ je neutrálním prvkem.

Spojením bezkontextové gramatiky a volné grupy generované totální abecedou této gramatiky získáme bezkontextovou gramatiku nad volnou grupou, kterou si nyní definujeme.

Tato problematika byla blíže popsána v [3].

Definice 4.1.1 *Bezkontextová gramatika nad volnou grupou* (zkráceně $\mathbf{CF}^\circ$) je čtveřice $\Gamma = (V, \Sigma, P, S)$, kde V, Σ a S má stejný význam jako v gramatikách bezkontextových. Dále P je konečná množina pravidel tvaru $A \to x$, kde $A \in V - \Sigma$ a $x \in V^*$.

Nás bude zajímat hlavně schopnost struktury generovat jazyky. Naším cílem je samozřejmě co nejvyšší síla — ideálně schopnost generovat celou třídu jazyků typu 0.

Ještě než začneme s podrobným průzkumem, musíme si definovat relaci přímé derivace $\Rightarrow^\circ$.

4.1.1 Derivace v bezkontextové gramatice nad volnou grupou

Definice 4.1.2 Nechť $\Gamma = (V, \Sigma, P, S)$ je bezkontextová gramatika nad volnou grupou. Mezi řetězci λ a μ platí relace $\Rightarrow^\circ$ zvaná přímá derivace, pokud oba řetězce můžeme vyjádřit ve tvaru

$$\lambda = \alpha A \beta$$

$$\mu = \alpha x \beta$$

a zároveň $p = A \rightarrow x \in P$, kde $\alpha, \beta, x \in V^*$. Potom píšeme $\lambda \Rightarrow^\circ_\Gamma \mu\ [p]$ a říkáme, že řetězec λ přímo derivuje řetězec μ podle pravidla p v bezkontextové gramatice nad volnou grupou Γ.

V případě bezprostředního výskytu dvojice $x\overline{x}$ nebo $\overline{x}x$, kde $x, \overline{x} \in V$, ve větné formě se tato okamžitě redukuje podle pravidel popsaných výše. Protože je výsledná redukovaná větná forma nezávislá na pořadí případných redukcí jednotlivých dvojic vzájemně inverzních symbolů, nečiní tento jev žádné problémy při generování vět jazyka.

Poznámka *Tranzitivní uzávěr* $\Rightarrow^{\circ+}_\Gamma$, *reflexivní a tranzitivní uzávěr* $\Rightarrow^{\circ*}_\Gamma$ a derivace délky n $\Rightarrow^{\circ n}_\Gamma$ jsou definovány zcela přirozeně stejným způsobem jako v případě běžných gramatik. Podobně symbol $_\Gamma$ jako dolní index symbolu zavedené relace nebudeme uvádět, pokud je z kontextu zřejmé, o kterou bezkontextovou gramatiku nad volnou grupou se jedná.

4.1.2 Jazyk generovaný bezkontextovou gramatikou nad volnou grupou

Definice 4.1.3 Nechť $\Gamma = (V, \Sigma, P, S)$ je bezkontextová gramatika nad volnou grupou. Řetězec $\alpha \in V^\circ$ nazýváme *větnou formou*, jestliže platí $S \Rightarrow^{\circ*} \alpha$, tj. řetězec α je generovatelný ze startovacího symbolu S. Větná forma, která obsahuje pouze terminální symboly, se nazývá *věta*. *Jazyk* $L(\Gamma)$, generovaný bezkontextovou gramatikou nad volnou grupou, je definován množinou všech vět

$$L(\Gamma) = \{w \mid S \Rightarrow^{\circ*} w \wedge w \in \Sigma^*\}$$

generovatelných z výchozího symbolu této gramatiky.

4.1.3 Generativní schopnosti bezkontextových gramatik nad volnými grupami

Nyní se budeme zabývat generativními schopnostmi nově zavedené struktury. Naším cílem je ukázat, že pro každou gramatiku H typu 0 existuje ekvivalentní bezkontextová gramatika nad volnou grupou Γ taková, že $L(H) = L(\Gamma)$.

Je tedy nutné najít určitý algoritmus transformace gramatik typu 0 na bezkontextové gramatiky nad volnými grupami.

Pravidla tvaru $A \rightarrow \varepsilon$, která mohou být aplikována kdekoliv, působí potíže při práci s volnou grupou (konkrétně se jedná o simulaci kontextových pravidel), proto bylo z [21] převzato následující lemma. To zajistí převedení pravidel tvaru $A \rightarrow \varepsilon$ na pravidla tvaru $A \rightarrow B$. Ke gramatice je poté přidána určitá režie, která zajistí přijímání původního jazyka. V této režii se vyskytuje pouze jedno pravidlo tvaru $A \rightarrow \varepsilon$, jeho použití je snadné ošetřit a ve výsledné gramatice nad volnou grupou nečiní jeho výskyt potíže. Více podrobností lze nalézt v příkladu konstrukce bezkontextové gramatiky nad volnou grupou. Nyní si uvedeme zmíněné lemma.

Lemma 4.1 Pro každou gramatiku $H = (N, \Sigma, P, S_H)$ typu 0 existuje ekvivalentní gramatika $G = (N \cup \{X, Y\}, \Sigma, P, S_G)$ typu 0, kde $\{X, Y, S_G\} \cap \{N \cup \Sigma\} = \emptyset$ taková, že její množina přepisovacích pravidel obsahuje pouze pravidla tvaru:

(1) $AB \to CD$ pro $A, B, C, D \in N$

(2) $A \to x$ pro $A \in N$ a $x \in N^2 \cup \Sigma$

(3) $AY \to YA$ pro $A \in N$

(4) $S_G \to XS_H$

(5) $XY \to X$

(6) $X \to \varepsilon$

Je zřejmé, že nová gramatika je v Kurodově normální formě a tedy její generativní síla bude odpovídat jazykům typu 0.

Důkaz 4.1.1 Z důkazu si uvedeme pouze konstrukci, kterou je též možné najít v [21]. Uvažujme gramatiku $H = (N_H, \Sigma, P_H, S_H)$. Bez ztráty na obecnosti můžeme předpokládat, že H je v Kurodově normální formě, tedy množina přepisovacích pravidel P_H obsahuje pouze pravidla tvaru

- $AB \to CD$
- $A \to BC$
- $A \to a$
- $A \to \epsilon$

$A, B, C \in N_H$, $a \in \Sigma$.

Navíc předpokládejme, že $\{S_G, X, Y\} \cap (N_H \cup \Sigma) = \emptyset$. Definujme novou gramatiku $G = (N_G, \Sigma, P_G, S_G)$, kde množina $N_G = N_H \cup \{X, Y\}$. Množinu přepisovacích pravidel P_G zkonstruujeme následovně:

(1) pokud $A \to x \in P_H$, kde $A \in N_H$ a $x \in N_H^2 \cup \Sigma$, potom přidej $A \to x$ do P_G

(2) pokud $AB \to CD \in P_H$, kde $A, B, C, D \in N_H$, potom přidej $AB \to CD$ do P_G

(3) pokud $A \to \varepsilon \in P_H$, kde $A \in N_H$, potom přidej $A \to Y$ do P_G

(4) pro každé $A \in N_H$ přidej $AY \to YA$ do P_G

(5) přidej $S_G \to XS_H$, $XY \to X$ a $X \to \varepsilon$ do P_G

Tím je konstrukce hotova.

Transformací gramatiky H na gramatiku G zajistíme, že se v nové gramatice G bude vyskytovat pouze jediné pravidlo typu $A \to \varepsilon$ a to právě pro $A = X$. Všechna pravidla z gramatiky H tvaru $A \to \varepsilon$ jsou v nové gramatice G převedena na pravidla $A \to Y$.

V gramatice G pomocí pravidel $AY \to YA$ zajistíme postupné předávání nonterminálu Y na začátek větné formy, respektive za nonterminál X. Po takovémto přesunu se uplatní pravidlo $XY \to X$. Tím je provedena simulace pravidel $A \to \varepsilon$ z původní gramatiky H.

Zbývá zavést na začátek každé větné formy nonterminál X. To zajistí pravidlo $S_G \to XS_H$. Po vymazání všech nonterminálů Y je zapotřebí odstranit i pomocný nonterminál X, k tomuto účelu slouží pravidlo $X \to \varepsilon$. Pro lepší pochopení je uveden následující příklad.

Poznámka Připomeňme, že podle zavedených konvencí z teorie formálních jazyků je třída všech jazyků typu 0 značena symbolem **RE**. Označme **CF**$^\circ$ třídu všech jazyků generovaných bezkontextovými gramatikami nad volnými grupami (ekvivalentní název by taktéž mohl být „třída všech jazyků generovaných gramatikami typu 2 nad volnými grupami”).

V tuto chvíli již máme definováno vše potřebné a můžeme si tedy vyslovit následující větu.

Věta 4.1.2 CF$^\circ$=RE

Jinými slovy tato věta říká, že pro každou gramatiku G typu 0 existuje ekvivalentní bezkontextová gramatika nad volnou grupou Γ taková, že $L(G) = L(\Gamma)$.

Důkaz 4.1.2 Z důkazu bude prezentována pouze konstrukce. Předpokládejme, že $G = (N_G, \Sigma, P_G, S)$ je gramatika typu 0 podle lemma 4.1.

Bezkontextovou gramatiku nad volnou grupou $\Gamma = (V, \Sigma, P_\Gamma, S)$, kde $N_\Gamma = N_G \cup N_{CS} \cup \overline{N_G}$, $V = N_\Gamma \cup \Sigma$ sestrojíme následovně:

I pokud $A \to x \in P_G$,

kde $x \in (N_G^2 \cup \{Y\} \cup \Sigma), A \in N_G$,

potom přidej $A \to x$ do P_Γ

II pokud $AB \to CD \in P_G$,

kde $A, B, C, D \in N_G$,

potom přidej $A \to C\langle ABCD\rangle$,

$B \to \overline{\langle ABCD\rangle}D$ do P_Γ

a $\langle ABCD\rangle, \overline{\langle ABCD\rangle}$ přidej do N_{CS}

III pro $XY \to X \in P_G$

přidej $X \to X\langle XYX\rangle$,

$Y \to \overline{\langle XYX\rangle}$ do P_Γ

a $\langle XYX\rangle, \overline{\langle XYX\rangle}$ přidej do N_{CS}

IV pro $X \to \varepsilon \in P_G$

přidej $X \to \langle X \rangle \langle \overline{X} \rangle$ do P_Γ

a $\langle X \rangle, \langle \overline{X} \rangle$ přidej do N_{CS}

V pro $Z \in N_G \cup \Sigma$

přidej $\overline{Z}$ do $\overline{N_G}$

Tím je konstrukce Γ hotova.

Množina $\overline{N_G}$ obsahuje inverzní symboly k symbolům z množiny $N_G \cup \Sigma$, tzn. které nejsou obsaženy v množině N_{CS}. Tato množina inverzních symbolů pouze doplňuje V na generátor volné grupy. Množina N_{CS} obsahuje nezbytně nutné nonterminály a jejich inverzní protějšky, které jsou potřeba pro další kroky důkazu a které se týkají kontextových přepisovacích pravidel.

Hlavní myšlenka důkazu Gramatika nad volnou grupou Γ simuluje aplikace kontextových pravidel tvaru $AB \to CD$ a to pomocí dvou derivačních kroků, které zajistí pravidla $A \to C\langle ABCD \rangle$ a $B \to \langle \overline{ABCD} \rangle D$ (bod **II**). Následující větná forma bude mít tvar $\alpha C \langle ABCD \rangle \langle \overline{ABCD} \rangle D\beta$. Pár $\langle ABCD \rangle \langle \overline{ABCD} \rangle$ je redukován pomocí vlastností volné grupy, tedy výsledná větná forma je $\alpha CD\beta$. Všimněme si, že symboly typu $\langle ABCD \rangle$ a $\langle \overline{ABCD} \rangle$ mohou být odstraněny z větné formy jen a pouze pomocí vlastnosti volné grupy, neexistuje totiž žádné pravidlo, které by tyto nonterminální symboly přepsalo. Je tedy zřejmé, že v případě nekorektního průběhu simulace původní gramatiky získáme tyto symboly bez svých inverzních protějšků, nedokážeme je eliminovat a zablokují tak derivaci. Ostatní pravidla, která mají tvar bezkontextových pravidel, jsou gramatikou Γ převzata.

Tento princip je stejný i u E0L gramatik nad volnými grupami.

Zbývá dokázat, že jsou obě gramatiky ekvivalentní, tedy že platí rovnost $L(G) = L(\Gamma)$. Nutně potom také $L(G) \subseteq L(\Gamma) \wedge L(\Gamma) \subseteq L(G)$.

4.2 E0L gramatiky nad volnými grupami

Druhá modifikace se bude týkat E0L gramatik. Konstrukce bude velmi podobná, proto bude popsána stručněji. Důvodem proč byly zvoleny E0L gramatiky, je jejich paralelní přístup k derivování větných forem a dále jsme jejich pomocí získali lepší výsledky v oblasti redukce nonterminálů.

Spojením E0L gramatiky a volné grupy generované totální abecedou této gramatiky získáme E0L gramatiku nad volnou grupou, kterou si nyní definujeme.

Tato problematika byla blíže popsána v [8].

Definice 4.2.4 *E0L gramatika nad volnou grupou* (zkráceně **E0L**$^\circ$) je čtveřice $\Gamma = (V, \Sigma, P, w)$, kde V a Σ má stejný význam jako v E0L gramatikách. Dále $w \in V^\circ$ a P je konečná množina pravidel tvaru $X \to x$, kde $X \in V$ a $x \in V^*$.

Pro úplnost definujeme relaci přímé derivace $\Rightarrow_\Gamma^\circ$.

4.2.1 Derivace v E0L gramatice nad volnou grupou

Definice 4.2.5 Nechť $\Gamma = (V, \Sigma, P, w)$ je E0L gramatika nad volnou grupou, kde $V = N \cup \Sigma$. Nechť λ a μ jsou řetězce z V°. Mezi řetězci λ a μ platí relace $\Rightarrow^\circ_\Gamma$, nazývaná *přímá derivace*, jestliže můžeme řetězce λ a μ vyjádřit ve tvaru:

$$\lambda = x_1 x_2 \dots x_n$$

$$\mu = y_1 y_2 \dots y_n$$

$$x_i \in V, \quad y_i \in V^\circ \text{ a } x_i \to y_i \in P \text{ pro } 1 \leq i \leq n.$$

Platí-li mezi řetězci λ a μ relace přímé derivace, pak píšeme $\lambda \Rightarrow^\circ_\Gamma \mu$ a říkáme, že řetězec μ lze přímo generovat z řetězce λ v E0L gramatice nad volnou grupou Γ.

Poznámka *Tranzitivní uzávěr* $\Rightarrow^{\circ+}_\Gamma$, *reflexivní a tranzitivní uzávěr* $\Rightarrow^{\circ*}_\Gamma$ a derivace délky n $\Rightarrow^{\circ n}_\Gamma$ jsou definovány zcela přirozeně stejným způsobem jako v případě E0L gramatik.

4.2.2 Jazyk generovaný E0L gramatikou nad volnou grupou

Definice 4.2.6 Nechť $\Gamma = (V, \Sigma, P, w)$ je E0L gramatika nad volnou grupou, kde $V = N \cup \Sigma$. Řetězec $\alpha \in V^\circ$ nazýváme *větnou formou*, jestliže platí $w \Rightarrow^{\circ*} \alpha$, tj. řetězec α je generovatelný z výchozího řetězce w. Větná forma, která obsahuje pouze terminální symboly, se nazývá *věta*. *Jazyk* $L(\Gamma)$, generovaný E0L gramatikou nad volnou grupou, je definován množinou všech vět

$$L(\Gamma) = \{u \mid w \Rightarrow^{\circ*} u, u \in \Sigma^*\}$$

generovatelných z výchozího řetězce této gramatiky.

4.2.3 Generativní schopnosti E0L gramatik nad volnými grupami

Nyní se budeme zabývat generativní schopností nově zavedené struktury. Cílem je dokázat, že pro každou gramatiku H typu 0 existuje ekvivalentní E0L gramatika nad volnou grupou Γ taková, že $L(H) = L(\Gamma)$.

Je tedy nutné opět sestrojit algoritmus transformace gramatik typu 0 na E0L gramatiky nad volnými grupami.

Poznámka Připomeňme, že podle zavedených konvencí z teorie formálních jazyků je třída všech jazyků typu 0 značena symbolem **RE**. Označme **E0L**$^\circ$ třídu všech jazyků generovaných E0L gramatikami nad volnými grupami.

V tuto chvíli již máme definováno vše potřebné a můžeme si tedy vyslovit následující větu.

Věta 4.2.3 **E0L**$^\circ$**=RE**

Jinými slovy tato věta říká, že pro každou gramatiku G typu 0 existuje ekvivalentní E0L gramatika nad volnou grupou Γ taková, že $L(G) = L(\Gamma)$.

Důkaz 4.2.3 Předpokládejme, že $G = (N_G, \Sigma, P_G, S)$ je gramatika typu 0 podle lemma 2.1.

E0L gramatiku nad volnou grupou $\Gamma = (V, \Sigma, P_\Gamma, w_\Gamma)$, kde $V = \Sigma \cup N_\Gamma$, $N_\Gamma = N_G \cup N_{CS} \cup \overline{N_G}$ a $w_\Gamma = S$ sestrojíme následovně:

I pokud $A \to x \in P_G$,
kde $x \in (N_G^2 \cup \{Y\} \cup \Sigma), A \in N_G$,
potom přidej $A \to x$ do P_Γ

II pokud $AB \to CD \in P$,
kde $A, B, C, D \in N_G$,
potom přidej $A \to C\langle ABCD \rangle$,
$B \to \langle \overline{ABCD} \rangle D$ do P_Γ
a $\langle ABCD \rangle, \langle \overline{ABCD} \rangle$ přidej do N_{CS}

III pro $XY \to X \in P_G$
přidej $X \to X\langle XYX \rangle$,
$Y \to \langle \overline{XYX} \rangle$ do P_Γ
a $\langle XYX \rangle, \langle \overline{XYX} \rangle$ přidej do N_{CS}

IV pro $X \to \varepsilon \in P_G$
přidej $X \to \langle X \rangle \langle \overline{X} \rangle$ do P_Γ
a $\langle X \rangle, \langle \overline{X} \rangle$ přidej do N_{CS}

V pro $Z \in N_G \cup \Sigma$
přidej $\overline{Z}$ do $\overline{N_G}$

VI pro každé $z \in V$
přidej $z \to z$ do P_Γ

Tím je konstrukce Γ hotova.

Množina $\overline{N_G}$ obsahuje inverzní symboly k symbolům z množiny $N_G \cup \Sigma$, tzn. které nejsou obsaženy v množině N_{CS}. Tato množina inverzních symbolů pouze doplňuje V na generátor volné grupy. Množina N_{CS} obsahuje nezbytně nutné nonterminály a jejich inverzní protějšky, které jsou potřeba pro další kroky důkazu a které se týkají kontextových přepisovacích pravidel.

Zbývá dokázat, že jsou obě gramatiky ekvivalentní, tedy že platí rovnost $L(G) = L(\Gamma)$. Nutně potom také $L(G) \subseteq L(\Gamma) \wedge L(\Gamma) \subseteq L(G)$.

Obě konstrukce jsou si velmi podobné (stejné až na bod **VI** a definici startovacího řetězce/nonterminálu), přesto se oba modely značně liší a to v definici přímé derivace. Bezkontextové gramatiky nad volnou grupou pracují sekvenčně a E0L gramatiky nad volnou grupou přepisují v každém kroku všechny symboly větné formy.

Kapitola 5

Gramatiky nad volnými grupami s redukovaným počtem nonterminálů

Při simulaci každého pravidla tvaru $AB \rightarrow CD$ vznikají jak v bezkontextových tak E0L gramatikách nad volnou grupou dva nové nonterminální symboly. Z tohoto důvodu se tato práce dále zajímá o redukci počtu nonterminálních symbolů v bezkontextových a E0L gramatikách nad volnou grupou.

5.1 Bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálů

V případě bezkontextových gramatik nad volnou grupou se podařilo dosáhnout redukce počtu nonterminálních symbolů na osm, jmenovitě to jsou, $0, \overline{0}, 1, \overline{1}, 2, \overline{2}, S$ a $\overline{S}$, přičemž nonterminální symbol $\overline{S}$ je zde z důvodů korektní definice volné grupy, nevyskytuje se v žádném z pravidel a tudíž ani v žádné větné formě. Síla těchto gramatik zůstává nezměněna a nadále popisují třídu jazyků typu 0.

V konstrukci budeme vycházet z upravené Kurodovy normální formy. Tuto úpravu definuje následující pomocná věta. Důvod použití této modifikace je blíže objasněn v příkladu konstrukce bezkontextové gramatiky nad volnou grupou s redukovaným počtem nonterminálů.

Tato problematika byla blíže popsána v [5].

Lemma 5.1 Pro každou gramatiku typu 0, $H = (V, T, P, S)$, $V = N \cup T$, existuje ekvivalentní gramatika typu 0, $G = (V_G, T, P_G, S)$, $V_G = N_G \cup T$, taková, že každé pravidlo z P_G je ve tvaru:

(1) $AB \rightarrow CD$, kde $A \neq C$ a $A, B, C, D \in N_G$

(2) $A \rightarrow BC$, kde $A \neq B$ a $A, B, C \in N_G$

(3) $A \rightarrow x$, kde $A \in N_G$, $x \in T \cup \{\varepsilon\}$

Důkaz 5.1.1 Nechť $H = (V, T, P, S)$ je gramatika. Bez ztráty na obecnosti předpokládejme, že gramatika H je v Kurodově normální formě.

Definujeme gramatiku $G = (V_G, T, P_G, S)$, $V_G = N_G \cup T$, kde P_G je vytvořena následovně:

(1) pro každé $AB \to AD \in P$ přidej $AB \to A'D'$, $A'D' \to AD$ do P_G a A', D' do N_G, kde A' a D' jsou dva nové nonterminály

(2) pro každé $A \to AB \in P$ přidej $A \to A'B'$, $A'B' \to AB$ do P_G a A', B' do N_G, kde A' a B' jsou dva nové nonterminály

(3) přidej všechna zbývající pravidla z P do P_G

Formální důkaz, že H a G jsou ekvivalentní, je snadný a proto je ponechán na čtenáři.

Poznámka Připomeňme, že pro třídu jazyků popsaných bezkontextovými gramatikami nad volnou grupou jsme si zavedli označení $\mathbf{CF}^\circ$. Třídu jazyků popsaných bezkontextovými gramatikami nad volnými grupami s redukovaným počtem nonterminálů budeme tedy označovat pomocí $\mathbf{CF}^\circ\mathbf{R}$.

V tuto chvíli již máme definováno vše potřebné a můžeme si tedy vyslovit následující větu.

Věta 5.1.1 $\mathbf{CF}^\circ\mathbf{R}$=**RE**

Důkaz 5.1.2 Uvažujme gramatiku typu 0, $G = (V, T, P, S)$, $V = N \cup T$. Bez ztráty na obecnosti lze předpokládat, že gramatika G splňuje vlastnosti popsané v lemma 5.1.

$\mathbf{CF}^\circ\mathbf{R}$ gramatiku, $\Gamma = (V_\Gamma, T, P_\Gamma, S_\Gamma)$, kde $V_\Gamma = N_\Gamma \cup T \cup \overline{T}$, $N_\Gamma = \{0, \overline{0}, 1, \overline{1}, 2, \overline{2}, S_\Gamma, \overline{S_\Gamma}\}$ sestrojíme následovně.

Připravíme si injekce $h : N \to \{0,1\}^{2*n}$ a $\overline{h} : N \to \{\overline{0}, \overline{1}\}^{2*n}$, kde $n = \lceil \log_2 |N| \rceil$ a pro které platí, že $h(A) = xy$ a $\overline{h}(A) = \overline{h(A)}$, kde $|x| = |y| = n$, $x = x_1 \ldots x_n$, $y = x_n \ldots x_1$, $x_1, \ldots, x_n \in \{0, 1\}$ pro každé $A \in N$.

Poznamenejme, že inverzní symboly k $0, 1, 2$ a $S_\Gamma \in N_\Gamma$ jsou $\overline{0}, \overline{1}, \overline{2}$ a $\overline{S_\Gamma} \in N_\Gamma$. Pomocí symbolů $0, 1, \overline{0}$ a $\overline{1}$ budeme kódovat nonterminály z původní množiny nonterminálních symbolů N a symboly $2, \overline{2}$ budou sloužit jako oddělovače kódů nonterminálů.

Vraťme se zpět ke konstrukci, nyní zbývá definovat množinu pravidel P_Γ:

I přidej $S_\Gamma \to h(S)2$ do P_Γ

II pro každé $AB \to CD \in P$ přidej $2 \to \overline{h}(B)\overline{2}\overline{h}(A)\overline{2}2h(C)2h(D)2$ do P_Γ

III pro každé $A \to BC \in P$ přidej $2 \to \overline{h}(A)\overline{2}2h(B)2h(C)2$ do P_Γ

IV pro každé $A \to x \in P$ přidej $2 \to \overline{h}(A)x$ do P_Γ

V pro každé $a \in T$ přidej $\overline{a}$ do $\overline{T}$

kde $A, B, C, D \in N$ a $x \in T \cup \{\varepsilon\}$. Poslední bod **V** je nutný k doplnění množiny V na generátor volné grupy, abeceda terminálních symbolů $\overline{T}$ se při generování vět jazyka nevyužije.

Konstrukce Γ je kompletní.

Hlavní myšlenka důkazu Gramatika nad volnou grupou Γ binárně kóduje nonterminály původní gramatiky G a simuluje aplikaci pravidel z původní gramatiky. Bezkontextové pravidla tvaru $A \rightarrow BC$ (bod **III**) jsou nahrazena pravidlem $\overline{h}(A)\overline{2}2h(B)2h(C)2$. Kde 2 je jediný nonterminál, který lze přepsat, tedy symbol 2 je přepsán a pokud se před ním nachází nonterminál A je vše korektní, protože tento symbol je odstraněn pomocí vlastností volné grupy a z aplikovaného pravidla tedy zbude pouze $h(B)2h(C)2$. Tento řetězec odpovídá BC. Na podobném způsobu je postavena simulace i ostatních pravidel. Vždy tedy platí, že se přepisuje pouze symbol 2, pokud tedy dojde k aplikaci nějakého pravidla na nesprávném místě, derivace se zablokuje, protože se neodstraní všechny nonterminální symboly.

Podobným způsobem lze popsat simulaci pomocí E0L gramatik nad volnou grupou s redukovaným počtem nonterminálních symbolů.

Zbývá dokázat, že obě gramatiky jsou ekvivalentní, tedy že platí rovnost $L(G) = L(\Gamma)$. Nutně potom také $L(G) \subseteq L(\Gamma)$ a $L(\Gamma) \subseteq L(G)$. Tato rovnost je dokázána v [2].

Důsledek 5.1.1 **CF°=CF°R**

Na závěr této podkapitoly poznamenejme, že způsob redukce využívající vlastností volných grup nikterak nezvýší počet pravidel vzhledem k původní gramatice v upravené Kurodově normální formě.

5.2 E0L gramatiky nad volnými grupami s redukovaným počtem nonterminálů

V této podkapitole popíšeme mechanismus převodu gramatiky typu 0 splňující výše popsané lemma 5.1 na E0L gramatiku nad volnou grupou s redukovaným počtem nonterminálů. Protože u E0L gramatik používáme místo startovacího symbolu startovací řetězec, dosáhneme tak lepšího výsledku než v případě bezkontextových gramatik nad volnou grupou. Výsledkem tedy je, že pro každou gramatiku typu 0 existuje E0L gramatika nad volnou grupou s pouze šesti nonterminálními symboly.

Konstrukce je opět velmi podobná, ale způsob provádění derivací je paralelní, narozdíl od bezkontextových gramatik nad volnou grupou, kde derivace probíhá sekvenčně.

Tato problematika byla blíže popsána v [9].

Poznámka Připomeňme, že pro třídu jazyků popsaných E0L gramatikami nad volnou grupou jsme si zavedli označení **E0L°**. Třídu jazyků popsaných E0L gramatikami nad volnými grupami s redukovaným počtem nonterminálů budeme tedy označovat pomocí **E0L°R**.

Věta 5.2.2 **E0L°R=RE**

Důkaz 5.2.3 Uvažujme gramatiku typu 0, $G = (V, T, P, S)$, $V = N \cup T$. Bez ztráty na obecnosti lze předpokládat, že gramatika G splňuje vlastnosti popsané v lemma 5.1.

E0L°R gramatiku, $\Gamma = (V_\Gamma, T, P_\Gamma, s_\Gamma)$, kde $V_\Gamma = N_\Gamma \cup T \cup \overline{T}$, $N_\Gamma = \{0, \overline{0}, 1, \overline{1}, 2, \overline{2}\}$ sestrojíme následovně.

Připravíme si opět injekce $h : N \to \{0,1\}^{2*n}$ a $\overline{h} : N \to \{\overline{0},\overline{1}\}^{2*n}$, kde $n = \lceil \log_2 |N| \rceil$ a pro které platí, že $h(A) = xy$ a $\overline{h}(A) = \overline{h(A)}$, kde $|x| = |y| = n$, $x = x_1 \dots x_n$, $y = x_n \dots x_1$, $x_1, \dots, x_n \in \{0,1\}$ pro každé $A \in N$.

Uveďme si pro lepší pochopení způsobu kódování malý příklad. Uvažujme $n = 3$ a například $h(A) = xy = 001100$, tedy $x = 001$, $x_1 = 0$, $x_2 = 0$ a $x_3 = 1$, podle definice injekce h musí být $y = x_3x_2x_1$ a tedy $y = 100$. Uvedený kód tedy splňuje požadované podmínky, podle kterých jsme tento kód nazvali zrcadlový.

Inverzní kód k $h(A)$ je následující, podle definice 5.3 inverzního řetězce se musí řetězec reverzovat (otočit) a každý symbol z řetězce je nahrazen svým inverzním protějškem. Kód $\overline{h}(A) = \overline{h(A)} = \overline{001100}$. Otočení se zde vzhledem k vlastnostem kódu žádným způsobem neprojevilo.

Poznámka: Aby následující text byl srozumitelnější, vyznačíme vybrané (delší) celky, které jsou vzájemně inverzní a jejichž výsledkem je prázdný řetězec ε, podtržením. Například výraz $h(x_{k+1})\underline{h(x_{k+1})\overline{h}(x_{k+1})}h(x_{k+1})$ je roven $h(x_{k+1})h(x_{k+1})$, tedy část zvýrazněná podtržením $\underline{h(x_{k+1})\overline{h}(x_{k+1})} = \varepsilon$.

Startovací řetězec s_Γ položíme rovno $h(S)2$. Nyní zbývá definovat množinu pravidel P_Γ:

I pro každé $z \in N_\Gamma \cup T$ přidej $z \to z$ do P_Γ

II pro každé $AB \to CD \in P$ přidej $2 \to \overline{h}(B)\overline{2}\overline{h}(A)\overline{2}2h(C)2h(D)2$ do P_Γ

III pro každé $A \to BC \in P$ přidej $2 \to \overline{h}(A)\overline{2}2h(B)2h(C)2$ do P_Γ

IV pro každé $A \to x \in P$ přidej $2 \to \overline{h}(A)x$ do P_Γ

V pro každé $a \in T$ přidej $\overline{a}$ do $\overline{T}$

kde $A, B, C, D \in N$ a $x \in T \cup \{\varepsilon\}$.

Konstrukce Γ je kompletní.

Hlavní myšlenka důkazu Princip simulace gramatiky typu 0 pomocí nově definované E0L gramatiky nad volnou grupou s redukovaným počtem nonterminálů je velice podobný simulaci pomocí **CF°R** gramatik, tento princip byl popsán v předchozí kapitole. Hlavním rozdílem oproti **CF°R** gramatikám je způsob provádění derivací. Rysem **E0L°R** gramatik je paralelní způsob provádění derivací. Ale z důvodu lepšího pochopení důkazu zavádíme v konstrukci pravidla tvaru $z \to z$, kde $z \in V_\Gamma$. Tento druh pravidel nám zajistí možnost uvažovat derivace v **E0L°R** gramatikách jako sekvenční. Přiblížíme se tedy velice blízko **CF°R** gramatikám.

Dokážeme, že obě gramatiky jsou ekvivalentní, tedy že platí rovnost $L(G) = L(\Gamma)$. Nutně potom také $L(G) \subseteq L(\Gamma)$ a $L(\Gamma) \subseteq L(G)$.

Důsledek 5.2.1 E0L°=E0L°R

Poznamenejme, že počet pravidel je shodný s počtem pravidel gramatiky typu 0, ze které tato konstrukce vychází.

Kapitola 6

Rozšířené oboustranné zásobníkové automaty nad volnými grupami

Poslední modifikovaný model je zásobníkový automat. Pomocí vlastností volné grupy se bude redukovat obsah zásobníku. Avšak výsledky získané přidáním volné grupy nikterak nezvýší generativní sílu samotných rozšířených oboustranných zásobníkových automatů nad volným monoidem, proč tomu tak je, bude ukázáno později. Samotné oboustranné zásobníkové automaty jsou definované v [2].

Tyto automaty se nazývají rozšířené, protože mají schopnost číst v jednom kroku ze vstupní pásky řetězce vstupní abecedy, - nerozšířené oboustranné zásobníkové automaty mohou číst v jednom kroku ze vstupní pásky pouze jeden vstupní symbol.

Nutno ještě poznamenat, že modifikací zásobníkových automatů, které přijímají třídu rekurzivně vyčíslitelných jazyků je nemalé množství, model který se nejvíce blíží modifikaci uvedené v této práci je tzv. Flip Pushdown Automaton, který je v podstatě také oboustranný, ale v jednom kroku lze pracovat vždy jen s jedním vrcholem (formální definici lze nalézt v [26]). U oboustranných zásobníkových automatů ale nejlépe využijeme vlastností volné grupy k redukci počtu pravidel a obsahu zásobníku.

Na začátek si zavedeme následující značení. Třídu všech jazyků přijímaných rozšířenými oboustrannými zásobníkovými automaty nad volnými grupami označíme **E2PDA**$^\circ$.

Tato kapitola byla ve velkém měřítku inspirovaná [19].

Definice 6.0.1 *Rozšířený oboustranný zásobníkový automat nad volnou grupou* je n-tice $M = (Q, \Sigma, \Gamma, R, z, Z_L, Z_R, F)$, $Q \cap (\Sigma \cup \Gamma) = \emptyset$, kde

- Q je konečná množina stavů
- Σ je konečná vstupní abeceda
- Γ je konečná zásobníková abeceda
- R je konečná množina pravidel tvaru $u_1|u_2pw \to v_1|v_2q$, kde $u_1, u_2 \in \Gamma$, $v_1, v_2 \in \Gamma^*$, $p, q \in Q$ a $w \in \Sigma^*$
- $z \in Q$ je počáteční stav
- $Z_L \in \Gamma$ je počáteční symbol levé strany zásobníku
- $Z_R \in \Gamma$ je počáteční symbol pravé strany zásobníku

- $F \subseteq Q$ je množina koncových stavů

6.1 Přechody v rozšířeném oboustranném zásobníkovém automatu nad volnou grupou

Definice 6.1.2 *Konfigurací* rozšířeného oboustranného zásobníku nad volnou grupou rozumíme řetězec vqy, kde $v \in \Gamma^\circ$, $y \in \Sigma^*$, and $q \in Q$.

Definice 6.1.3 Pokud $u_1|u_2qw \rightarrow v_1|v_2p \in R$, $y = u_1hu_2qwz$ a $x = v_1hv_2pz$ jsou dvě konfigurace automatu M, kde $u_1, u_2 \in \Gamma$, $x, y \in \Gamma^\circ$, $q, p \in Q$ a $w, z \in \Sigma^*$, potom automat M provádí *přechod* z konfigurace y do konfigurace x podle pravidla $u_1|u_2qw \rightarrow v_1|v_2p$ a píšeme

$$y \vdash_M^\circ x[u_1|u_2qw \rightarrow v_1|v_2q]$$

nebo jednoduše

$$y \vdash^\circ x$$

Relace $\vdash^{\circ n}$, $\vdash^{\circ +}$ a $\vdash^{\circ *}$ označující posloupnost délky n, pro $n \geq 0$, tranzitivní a reflexivní-tranzitivní uzávěr relace $\vdash^\circ$ v tomto pořadí jsou definovány obvyklým způsobem.

6.2 Jazyk generovaný rozšířeným oboustranným zásobníkovým automatem nad volnou grupou

Definice 6.2.3 Jazyk přijímaný rozšířeným zásobníkovým automatem nad volnou grupou je definován jako

$$L(M) = \{w|w \in \Sigma^*, Z_LZ_Rzw \vdash^{\circ *} \varepsilon f, f \in F\}.$$

Poznamenejme, že řetězce vyskytující se na oboustranném zásobníku jsou tvořeny volnou grupou generovanou zásobníkovou abecedou Γ operací konkatenace. Řetězec ω je automatem M přijat pouze tehdy, pokud je celý přečten, zásobník je prázdný a automat M se nachází v některém z koncových stavů.

6.3 Generativní schopnosti rozšířených oboustranných zásobníkových automatů nad volnou grupou

Nyní se budeme zabývat generativní schopností nově zavedené struktury. Cílem je dokázat, že pro každou zleva rozšířenou gramatiku G existuje rozšířený oboustranný zásobníkový automat nad volnou grupou M přijímající stejný jazyk, $L(G) = L(M)$.

Věta 6.3.1 E2PDA$^\circ$=RE

Důkaz 6.3.1 Je dokázáno, že třída jazyků generovaných zleva rozšířenými frontovými gramatikami (viz kapitola 2.2) je totožná s třídou rekurzivně vyčíslitelných jazyků. Postačuje tedy dokázat, že pro každou zleva rozšířenou frontovou gramatiku $G = (V, T, W, F, Sq_0, P)$ existuje rozšířený oboustranný zásobníkový automat nad volnou grupou $M = (Q, T, \Gamma, R, z, Z_L, Z_R, F)$ takový, že $L(G) = L(M)$. Bez ztráty na obecnosti předpokládejme, že gramatika G splňuje podmínky popsané v lemma 2.1.

Konstrukci rozšířeného oboustranného zásobníkového automatu nad volnou grupou M, která byla prezentována v [10], provedeme aplikací následujících kroků:

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

Množina pravidel R je vytvořena následovně:

I pro startovací 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

II pro každé $(A, p, x, q) \in P$, kde $A \in (V - T)$, $p, q \in (W - F)$, $x \in (V - T)^*$
přidej $Z_L | Z_R \langle p, 1\rangle \to Z_L \overline{A} | x Z_R \langle q, 1\rangle$ do R

III pro každé $p \in W$
přidej $Z_L | Z_R \langle p, 1\rangle \to Z_L | Z_R \langle p, 2\rangle$ do R

IV pro každé $(A, p, y, q) \in P$, kde $A \in (V - T)$, $p, q \in (W - F)$, $y \in T^*$
přidej $Z_L | Z_R \langle p, 2\rangle y \to Z_L \overline{A} | Z_R \langle q, 2\rangle$ do R

V pro každé $(A, p, y, t) \in P$, kde $A \in (V - T)$, $p \in (W - F)$, $y \in T^*$, $t \in F_M$
přidej $Z_L | Z_R \langle p, 2\rangle y \to \overline{A} | \varepsilon f$ do R

Pokud si uvědomíme jakým způsobem se využívá vlastností volné grupy - redukují se dvojice nonterminálů $\overline{A}A$, dokud zásobník není prázdný - lze tento mechanismus nahradit pravidly tvaru $A' | Ap \to \varepsilon | \varepsilon q$, tyto pravidla budou aplikována po načtení celého vstupního řetězce. Budou pravděpodobně vyžadovat další zásobníkový stav. Takto upravená verze bez volné grupy bude mít stejnou vyjadřovací schopnost jako výše definovaná verze využívající volnou grupu.

V tomto případě výhodu použití volných grup lze spatřit v redukci počtu pravidel, stavů a automatického mazání symbolů uvnitř zásobníku, čímž se šetří paměťové nároky v případě dynamické alokace.

Kapitola 7

Rozšířené oboustranné zásobníkové automaty nad volnými grupami s redukovanou zásobníkovou abecedou

Způsob kódování nonterminálních symbolů, který jsme zavedli při redukci nonterminální abecedy bezkontextových gramatik a E0L gramatik, lze využít i v případě kódování zásobní-kové abecedy oboustranných zásobníkových automatů. Dosáhneme tak velice dobrého výsle-dku a to redukci zásobníkové abecedy přesně na čtyři symboly, konkrétně se jedná o symboly $0, \overline{0}, 1$ a $\overline{1}$.

Než přistoupíme k samotnému důkazu generativní síly těchto automatů, zavedeme si následující značení. Rozšířené zásobníkové automaty nad volnou grupou s redukovanou zásobníkovou abecedou bude značit jako **E2PDA°R**.

Věta 7.0.1 E2PDA°R=RE

Důkaz 7.0.1 Je dokázáno, že třída jazyků generovaných zleva rozšířenými frontovými gramatikami (viz kapitola 2.2) je totožná s třídou rekurzivně vyčíslitelných jazyků. Postačuje tedy dokázat, že pro každou zleva rozšířenou frontovou gramatiku $G = (V, T, W, F, Sq_0, P)$ existuje rozšířený oboustranný zásobníkový automat nad volnou grupou $M = (Q, T, Z, R, z, 1, 1, F_M)$ takový, že $L(G) = L(M)$. Bez ztráty na obecnosti předpokládejme, že gramatika G splňuje podmínky popsané v lemma 2.1.

Konstrukce rozšířeného zásobníkového automatu nad volnou grupou s redukovanou zásobníkovou abecedou je následující.

Zavedeme injekce $h : (V - T) \to \{0, 1\}^{n+2}$ a $\overline{h} : (V - T) \to \{\overline{0}, \overline{1}\}^{n+2}$, kde $n = \lceil \log_2(card(V - T)) \rceil$, takové že pro každé $A \in (V - T)$, $h(A) = \{0\}\{0, 1\}^n\{0\}$ a $\overline{h}(A) = \overline{h(A)}$. Rozšíříme doménu h na $(V - T)^*$. Nyní je h injektivní homomorfismus z $(V - T)^*$ na $(\{0\}\{0, 1\}^n\{0\})^*$. Poznamenejme, že inverzní symboly k 0 a $1 \in V - T$ jsou $\overline{0}$ a $\overline{1} \in V - T$.

- $Q = \{f, z\} \cup \{\langle q, 1\rangle, \langle q, 2\rangle | q \in W\}$
- $Z = \{0, \overline{0}, 1, \overline{1}\}$
- $F_M = \{f\}$

Množina pravidel R je vytvořena následovně:

I pro startovací axiom Sq_0 gramatiky G, kde $S \in (V-T)$, $q_0 \in (W-F)$
přidej $1|1z \rightarrow 1|h(S)1\langle q_0, 1\rangle$ do R

II pro každé $(A, q, x, p) \in P$, kde $A \in (V-T)$, $p, q \in (W-F)$, $x \in (V-T)^*$
přidej $1|1\langle q, 1\rangle \rightarrow 1\overline{h}(A)|h(x)1\langle p, 1\rangle$ do R

III pro každé $q \in W$
přidej $1|1\langle q, 1\rangle \rightarrow 1|1\langle q, 2\rangle$ do R

IV pro každé $(A, q, y, p) \in P$, kde $A \in (V-T)$, $p, q \in (W-F)$, $y \in T^*$
přidej $1|1\langle q, 2\rangle y \rightarrow 1\overline{h}(A)|1\langle p, 2\rangle$ do R

V pro každé $(A, q, y, t) \in P$, kde $A \in (V-T)$, $q \in (W-F)$, $y \in T^*$, $t \in F$
přidej $1|1\langle q, 2\rangle y \rightarrow \overline{h}(A)|\varepsilon f$ do R

Konstrukce automatu M je kompletní.

Hlavní myšlenka důkazu Automat M simuluje derivace v zleva rozšířené frontové gramatice G a ukládá na zásobník binárně kódované symboly z $V-T$. Uvažujme jako aktuální větnou formu $w\#Avp$ gramatiky G, kde w, $v \in (V-T)^*$, $A \in (V-T)$, a $p \in (W-F)$. Potom odpovídající konfigurace automatu M je $1\overline{h}(w)h(w)h(A)h(v)1\langle p, 1\rangle\omega$, kde $\omega \in T^*$. Nechť existuje nějaké pravidlo $(A, p, x, q) \in P$, kde $x \in (V-T)^*$, potom $w\#Avp \Rightarrow wA\#vxq$ v G. V takovém případě, automat M musí být v *nonterminal-generating* módu a podle konstrukce obsahuje pravidlo $1|1\langle p, 1\rangle \rightarrow 1\overline{h}(A)|h(x)1\langle q, 1\rangle \in R$. Použitím tohoto pravidla přechází automat M do nové konfigurace $1\overline{h}(A)\overline{h}(w)h(w)h(A)h(v)h(x)1\langle q, 1\rangle\omega$. Všimněme si, že symbol A je kódován jako $\overline{h}(A)$ a výsledný binární kód tohoto symbolu je vložen z levé strany oboustranného zásobníku. Dále je řetězec x kódován pomocí injekce h a výsledný binární kód je vložen do oboustranného zásobníku z pravé strany.

Dále uvažujme jako aktuální větnou formu $w\#Avup$ gramatiky G, kde $u \in T^*$ a $(A, p, y, q) \in P$, $y \in T^*$. Potom lze provést $w\#Avup \Rightarrow wA\#vuyq$. Podle konstrukce musí existovat v automatu M pravidlo $1|1\langle p, 2\rangle y \rightarrow 1\overline{h}(A)|1\langle q, 2\rangle \in R$ a automat M může provést přechod $1\overline{h}(w)h(w)h(A)h(v)1\langle p, 2\rangle y\omega' \vdash^\circ 1\overline{h}(A)\overline{h}(w)h(w)h(A)h(v)1\langle q, 2\rangle\omega'$, kde $\omega' \in T^*$. Poznamenejme, že v tomto případě automat M musí být v módu *terminal-reading*. V tomto režimu jsou binárně kódovány pouze symboly A. Na levou stranu oboustranného zásobníku se tedy vkládá binární řetězec $\overline{h}(A)$.

Jinými slovy, každý symbol $A \in (V-T)$, který je generován za $\#$ v gramatice G, je do oboustranného zásobníku vložen z pravé strany jako $h(A)$. Všimněme si, že každý takovýto symbol ve frontové gramatice G splňující lemma 2.1 je přesunut před $\#$. Z tohoto důvodu ukládá automat M inverzní protějšky těchto symbolů na levou stranu oboustranného zásobníku. Pokud se všechny přechody provedou korektně, oboustranný zásobník se pomocí vlastností volné grupy sám vyprázdní.

Kapitola 8

Praktické využití

Přestože tato práce přináší nové výsledky na poli teoretické informatiky, pokusíme se nyní nastínit možné využití jejich výsledků i v praktické oblasti.

8.1 Překladače

Hlavní oblastí využití formálních modelů nad volnými grupami by se měla stát oblast překladačů. Většina současných překladačů je založena na bezkontextových gramatikách popisujících bezkontextový jazyk. Bezkontextová gramatika nad volnou grupou zachovává tvar bezkontextových pravidel, ale dokáže popsat o dvě třídy vyšší jazyk — rekurzivně vyčíslitelný jazyk, může se tedy stát zajímavým řešením pro syntaktický analyzátor pracující shora dolů.

Kromě gramatiky lze pro kompilátor použít také zásobníkový automat, který je vhodný pro syntaktický analyzátor pracující metodou zdola nahoru. Jednou ze studií, jak využít některý zde nově zavedený formální model nad volnou grupou, je diplomová práce [1], ve které je navržen a implementován překladač založený na zásobníkovém automatu nad volnou grupou. Navržený zásobníkový automat je ale nedeterministický a časově tedy značně náročný. Tento aspekt je jednou z nevýhod syntaktické analýzy pracující zdola nahoru, je to způsobeno tím, že prázdný řetězec může být kdekoliv a kdykoliv rozšířen na řetězec složený ze vzájemně inverzních nonterminálních symbolů. Praktické využití pro překladače založené na syntaktické analýze zdola nahoru by tedy znamenalo přidání nějaké další režie pro lepší predikci kdy a kde rozšířit prázdný řetězec.

8.2 Bezpečnostní kód

V této práci definovaný "zrcadlový kód" by mohl nalézt využití i v jiných oblastech informačních technologií. Například by se mohlo jednat o chybu-detekující kód. Jednalo by se o kód, který se dá nejlépe přirovnat k tzv. zdvojenému kódu.

Zdvojený kód využívá k detekci chyby zdvojení, což znamená, že každý bit je reprezentován dvěma stejnými bity, například pro kód $\ldots 101 \ldots$ by byl zdvojený kód $\ldots 110011 \ldots$. V případě chyby $\ldots \underline{10}0011 \ldots$ je kód detekován jako poškozený. Podobně by se pro kód $\ldots 101 \ldots$ vytvořil zrcadlový kód $\ldots 101101 \ldots$ a v případě chyby $\ldots \underline{0}0110\underline{1}$ je kód opět detekován jako chybný.

Pokud budeme uvažovat možnost, že v kódu může dojít ke ztrátě informace, jeden nebo více bitů je při přenosu ztraceno, zdvojený kód je schopen detekovat takovouto chybu pouze při ztrátě lichého počtu bitů. Například z $\ldots 110011 \ldots$ na $\ldots 11\underline{0}11 \ldots$ je chyba rozpoznána.

V případě přenosu $\dots 110011 \dots$ na $\dots 1100 \dots$ došlo ke ztrátě 11, ale zdvojený kód tuto chybu nedetekuje. Zrcadlový kód dokáže zjistit chybu způsobenou ztrátou informace lépe. Kontroluje se zda je každé slovo vnitřně reverzní, tzn. že je tvaru $a_1 a_2 \dots a_n a_n \dots a_2 a_1$ a má definovanou délku n. Zrcadlový kód tedy v tomto případě má lepší vlastnosti než kód zdvojený.

8.3 Expertní simulace

Samotný proces zkracování řetězců pomocí volné grupy by mohl najít uplatnění ve vybraných situacích, kde při sloučení, spojení nebo nalezení určitých prvků dojde k jejich neutralizaci nebo eliminaci. Přitom nezáleží na jejich pořadí. To znamená, že pokud bude například připojen řetězec x k řetězci $\overline{x}$ v nějaké situaci popsané jako přechod z $\dots xA \dots$ na $\dots x\overline{x} \dots = \dots \varepsilon \dots$, tak stejný výsledek bude mít situace při přechodu z $\dots \overline{x}A \dots$ na $\dots \overline{x}x \dots = \dots \varepsilon \dots$.

Konkrétním příkladem může být zavedení volné grupy do speciálních L-systémů prezentovaných v [23]. V této publikaci jsou představeny nové L-systémy rozšířené o řízení pomocí zakazujících nebo povolujících podmínek. Aplikace těchto systémů pak může být v biologii, kde jsou tyto systémy schopny popisovat například situaci napadení organické buňky virem, její růst, uzdravení nebo i smrt. Logickým rozšířením těchto systémů mohou být E0L gramatiky nad volnými grupami, které by mohli lépe popsat proces léčení, kdy organická buňka získá určitou léčebnou látku, která s reakcí prvku představujícího onemocnění zmizí i s tímto prvkem. Buňka pak tedy přijde o stejný počet léčebných látek tak i stejný počet prvků představujících onemocnění, její výsledný stav bude záviset na tom, jaké prvky v buňce zbyly. Tento příklad je pouze hrubý náčrt, při praktickém využití by celý proces závisel na konkrétním organismu a možném onemocnění.

Kapitola 9

Závěr

V tomto dokumentu byly modifikovány některé vybrané více či méně známé formální modely. Tyto modifikace převážně spočívaly v zavedení relace přímé derivace nad volnými grupami místo nad volnými monoidy, jak je to v teoretické informatice běžné. Motivací pro tuto změnu je snaha o zvýšení vyjadřovací síly takto upravených formálních modelů. V případě modelů založených na gramatikách se podařilo dokázat, že touto modifikací opravdu vyjadřovací sílu zvýšíme a přitom zachováme tvar přepisovacích pravidel, který je definovaný v originálních gramatikách. Získali jsme tedy nové formální modely $\mathbf{CF}^{\circ}$, $\mathbf{E0L}^{\circ}$, $\mathbf{CF}^{\circ}\mathbf{R}$ a $\mathbf{E0L}^{\circ}\mathbf{R}$, které využívají pouze bezkontextových přepisovacích pravidel (tzn. přepisovacích pravidel, která mají na levé straně pouze jediný symbol) a popisují tak třídu rekurzivně vyčíslitelných jazyků.

Neméně významnou částí je také snaha o redukci počtu nonterminálních symbolů v těchto nově zavedených formálních modelech, přičemž samozřejmě zachováváme tvar přepi-sovacích pravidel i vyjadřovací sílu tohoto modelu. Tato problematika byla například velice dobře popsána v [**?**] a bylo zde dosaženo i velice dobrých výsledků (množina nonterminálních symbolů byla redukována pouze na dva páry závorek, ale k jejich odstranění je zapotřebí kontextových pravidel). V této práci je představena $\mathbf{E0L}^{\circ}\mathbf{R}$ gramatika, která k popisu rekurzivně vyčíslitelného jazyka potřebuje pouze šest nonterminálních symbolů, nebo v případě automatů byl představen $\mathbf{E2PDA}^{\circ}\mathbf{R}$ automat, který dokáže přijímat rekurzivně vyčíslitelný jazyk pouze za pomoci čtyř nonterminálních symbolů, které tvoří celou zásobní-kovou abecedu.

Tato práce je pouze prvním krokem v oblasti formálních modelů nad volnými grupami, její pokračování by se mělo převážně zabývat studií o časové a paměťové náročnosti a porovnat ji s již známými formálními modely. Důležitou částí budoucího výzkumu se může stát i studie zaměřená na deterministické verze zde nově popsaných gramatik a automatů, zvláště otevřený problém je, zda bude zachována síla odpovídající nedeterministickým verzím. Pokud výsledky v těchto oblastech budou příznivé, měli by být velmi dobrou motivací pro nalezení praktického využití těchto nových formálních modelů, které jsou v tomto okamžiku výborným výsledkem převážně v oblasti teoretické.

Nyní si provedeme zběžnou rekapitulaci dosažených výsledků této práce. V případě gramatik byly zkoumány dva pohledy vzhledem k derivacím:

- *Sekvenční přístup:* byla definována bezkontextová gramatika nad volnou grupou, která v každém derivačním kroku přepisuje pouze jediný nonterminální symbol, derivace jsou tedy prováděny sekvenčně. Bylo dokázáno, že tento nově zavedený formální model má stejnou sílu jako Turingův stroj a popisuje tedy třídu rekurzivně vyčíslitelných jazyků,

$\mathbf{CF}^\circ = \mathbf{RE}$. Požadavky na gramatiku typu 0, podle které lze $\mathbf{CF}^\circ$ gramatiku sestrojit i samotná konstrukce je ukázána na příkladě.

Dále byla zkoumána možnost redukce počtu nonterminálních symbolů a získali jsme jako výsledek, že každý rekurzivně vyčíslitelný jazyk lze popsat pomocí bezkontextové gramatiky nad volnou grupou pouze s osmi nonterminálními symboly, $\mathbf{CF}^\circ\mathbf{R} = \mathbf{RE}$, přičemž při všech možných derivačních krocích se využije pouze sedm nonterminálních symbolů, jeden z nich musí být zadefinován pouze z důvodu, aby totální abeceda této gramatiky mohla být použita jako generátor volné grupy. Konstrukce této redukované verze je opět ilustrována na příkladě.

- *Paralelní přístup:* jako další formální model vybraný k modifikaci byla zvolena E0L gramatika, její pravidla jsou podobná bezkontextové gramatice, ale derivace lze prová-dět paralelně, to znamená, že v každém derivačním kroku se musí přepsat všechny nonterminální symboly v dané větné formě. Připomeňme si, že pokud jeden nebo více symbolů nelze přepsat a větná forma je složena pouze z terminálů, derivace je korektně ukončena. Pokud nelze provést derivaci u jednoho nebo více symbolů a větná forma obsahuje jeden nebo více nonterminálních symbolů derivace je zablokována.

  Byla definována E0L gramatika nad volnou grupou a bylo dokázáno, že tato gramatika má stejnou vyjadřovací schopnost jako Turingův stroj, stejně jako $\mathbf{CF}^\circ$ tedy tato gramatika popisuje třídu rekurzivně vyčíslitelných jazyků, $\mathbf{E0L}^\circ = \mathbf{RE}$.

  Stejně jako u bezkontextových gramatik nad volnou grupou tak i v tomto případě se práce zabývá redukcí počtu nonterminálních symbolů. Bylo dokázáno, že každý rekurzivně vyčíslitelný jazyk lze popsat pomocí E0L gramatiky nad volnou grupou, která má pouze šest nonterminálních symbolů, $\mathbf{E0L}^\circ\mathbf{R} = \mathbf{RE}$. Obě konstrukce jsou pro lepší pochopení opět doprovázeny příklady.

Pokud uvážíme různé generativní síly tříd bezkontextových a E0L gramatik, $\mathbf{CF} \subset \mathbf{E0L}$, je také zajímavým výsledkem stejná síla gramatik bezkontextových nad volnou grupou a E0L gramatik nad volnou grupou, $\mathbf{CF}^\circ = \mathbf{E0L}^\circ$.

Tato tématika byla publikována v časopisecké publikaci [7], na několika mezinárodních konferencích [4], [3] a [8] a dále také na národních konferencích [5], [9].

Zbylá část této práce se zabývá využitím volných grup u zásobníkového automatu. Byly definovány rozšířené oboustranné zásobníkové automaty nad volnou grupou. Tato modifikace nepřináší zvýšení síly, protože už samotné rozšířené oboustranné zásobníkové automaty nad volným monoidem mají sílu Turingových strojů, což bylo dokázáno v [2].

Běžný rozšířený zásobníkový automat byl modifikován tak, aby bylo možné vkládat a odebírat symboly z obou jeho stran, tato konstrukce dovoluje i více transformací. Například pokud povolíme vkládání symbolů jen z jedné strany a odebírání ze strany druhé, získáme tak frontu. Dále pak v tomto případě záměna volného monoidu za volnou grupu přináší redukci počtu pravidel. Nejsou totiž potřeba žádná odebírací pravidla. Při správném výpočtu se oboustranný zásobník pomocí vlastností volné grupy sám zkracuje v průběhu své činnosti, je tedy dosaženo menší paměťové náročnosti v porovnání s oboustranným rozšířeným zásobníkovým automatem nad volným monoidem popsaném v [2]. Bylo tedy dokázáno, že síla rozšířeného oboustranného zásobníkového automatu nad volnou grupou zůstává rovna síle Turingova stroje, $\mathbf{E2PDA}^\circ = \mathbf{RE}$.

I v případě zásobníkových automatů se tato práce zabývala redukcí počtu nonterminálních symbolů, konkrétně se jednalo o redukci počtu nonterminálních symbolů zásobní-kové abecedy. Bylo dokázáno, že k popisu každého rekurzivně vyčíslitelného jazyka postačuje oboustranný rozšířený zásobníkový automat nad volnou grupou pouze se čtyřmi symboly zásobníkové abecedy, $\mathbf{E2PDA}^{\circ}\mathbf{R} = \mathbf{RE}$.

Tato problematika byla publikována v časopisecké publikaci [6], na mezinárodní konferenci [4] a dále také na národní konferenci [10].

Zhodnocení výše uvedených výsledků můžeme rozdělit do dvou skupin:

- *Teoretický přínos:* je zřejmý, v případě gramatik nad volnými grupami jsme získali formální modely, které používají pouze pravidla bezkontextového tvaru, $A \rightarrow x$, ale jejich vyjadřovací síla odpovídá síle Turingova stroje, navíc je této síly u gramatik dosaženo i s redukovaným počtem nonterminálních symbolů nebo u oboustranného automatu s redukovanou zásobníkovou abecedou. Tento teoretický přínos byl také potvrzen přijetím dvou článků, které pojednávají o $\mathbf{CF}^{\circ}\mathbf{R}$, $\mathbf{E0L}^{\circ}\mathbf{R}$ a $\mathbf{E2PDA}^{\circ}\mathbf{R}$, do mezinárodních časopisů, konkrétně se jedná o [7] a [6].

- *Praktický přínos:* byla zkoumána možnost použit některý z navržených modelů pro syntaktický analyzátor. V diplomové práci Ing. Michala Berky, [1], byl úspěšně implementován syntaktický analyzátor založený na zásobníkovém automatu nad volnou grupou. V této práci byl zaveden pojem expanze, který na libovolném místě rozšiřuje ε na $x\overline{x}$ nebo $\overline{x}x$, kde x může být terminální symbol, nonterminální symbol nebo i řetězec.

Literatura

[1] Berka, M.: Zásobníkový automat nad volnou grupou, *Diplomová práce*, Vysoké učení technické v Brně, Fakulta informačních technologií, 2005.

[2] Bidlo, R.: O síle některých modifikovaných formálních modelů, *Disertační práce*, Vysoké učení technické v Brně, Fakulta informačních technologií, 2007, (poddáno k oponentuře).

[3] Bidlo, R.: Context-Free Grammars over Free Groups, *Proceedings of 8th International Conference* ISIM'05, 2005, p. 95-100.

[4] Bidlo, R., Blatný, P., Meduna, A.: Formal Models over Free Groups. *PRE-PROCEEDINGS of the 1st Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, 2005, p. 193-199.

[5] Bidlo, R., Blatný, P. How to Generate Recursively Enumerable Languages Using Only Context-free Productions and Eight Nonterminals, *Proceedings of 11th Conference and Competition Student* EEICT 2005, Vol. *3*, 2005, p. 536-541.

[6] Bidlo, R., Blatný, P., Meduna, A.: Automata with Two-Sided Pushdowns Defined over Free Groups Generated by Reduced Alphabets, Kybernetika, Vol. *2007*, No. 1, 2007, p. 21-35.

[7] Bidlo, R., Blatný, P., Meduna, A.: Context-Free and E0L Derivations over Free Groups, Schedae Informaticae, Krakov, 2007, p. 14-24.

[8] Blatný, P.: E0L Grammars on Free Groups, *Proceedings of 8th International Conference* ISIM'05, 2005, p. 81-86.

[9] Blatný, P., Bidlo, R.: The Parallel Generation of Recursively Enumerable Languages Using Only Context-free Productions and Six Nonterminals, *Proceedings of 11th Conference and Competition Student* EEICT 2005, Vol. *3*, 2005, p. 541-546.

[10] Blatný, P., Bidlo, R.: Two-Sided Pushdown Automata Over Free Groups, *Proceedings of 12th Conference and Competition Student* EEICT 2006, Vol. *3*, 2006.

[11] Dassow, J., Mitrana, V.: Finite Automata Over Free Groups. *Int. Journalof Algebra and Computation* Vol. *10*, No. *6*, 2000, p. 725-737.

[12] Drápal, A.: Teorie grup - základní aspekty. Karolinum, Praha, 2000.

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

[14] Kleijn, H. C. M., Rozemberg, G.: On the Generative Power of Regular Pattern Grammars. *Acta Informatica 20*, 1983, p. 391-411.

[15] Kuroda, S.Y.: Classes of Languages and Linear Bounded Automata, *Inform. Control 7*, 1964, p. 207-223.

[16] Kuroš, A., G.: Kapitoly z obecné algebry. Academia, Praha, 1977.

[17] Lindenmayer, A.: Mathematical models of cellular interaction in development I and II, *J. Theoret. Biol. 54*, 1975, p. 3-22.

[18] Meduna, A.: Symbiotic E0L systems, *Artificial Life: Gramatical Models*, Bucharest, 1995, p. 122-129.

[19] Meduna, A.: Simultaneosly One-Turn Two-Pushdown Automata. *International Journal of Computer Mathematics 82*, Taylor & Francis Informa plc, 2003, p. 1-9.

[20] Meduna, A.: Context Free Derivations on Word Monoids. *Acta Informatica 27*, Springer-Verlag, 1990, p. 781-786.

[21] Meduna, A., Kolář, D.: Homogenous Grammars with a Reduced Number of Non-Context-Free Productions. *Information Processing Letters 81*, Amsterdam, 2002, p. 253-257.

[22] Meduna, A., Kolář, D.: Regulated Pushdown Automata. *Acta Cybernetica 14*, 2000, p. 653-664.

[23] Meduna, A., Švec, M.: Grammars with Context Conditions and Their Applications, Wiley, Hoboken, New Jersey, 2005.

[24] Mitrana, V., Stiebe, R.: The Accepting Power of Finite Automata Over Groups. *TUCS Technical report 70*, 1996.

[25] EDs.: Rozenberg, G., Saloma, A.: Hangbook of Formal Languages, Vol. *1*, Springer, 1997.

[26] Sarkar, P.: Pushdown Automaton with Ability to Flip its Stack. *ECCC*, No. *81*, 2001, p. 1-6.