VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

O SÍLE NĚKTERÝCH MODIFIKOVANÝCH FORMÁLNÍCH MODELŮ

ON THE POWER OF SOME MODIFIED FORMAL MODELS

TEZE DIZERTAČNÍ PRÁCE

SHORT PHD THESIS

AUTOR PRÁCE — Ing. RADEK BIDLO

AUTHOR

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

SUPERVISOR

BRNO 2007

Abstrakt

Předložená dizertační práce se zabývá problematikou modifikace různých formálních modelů a studiem dopadu těchto modifikací na jejich vyjadřovací schopnosti. Celkem jsou zkoumány tři nové formální modely. Z oblasti automatů je zaveden oboustranný zásobníkový automat a jeho konstrukce s využitím frontových gramatik. Studována je i verze s redukovaným počtem symbolů zásobníkové abecedy. V obou těchto případech je použitím oboustranného zásobníku zvýšena vyjadřovací síla běžných zásobníkových automatů až na úroveň Turingova stroje. Dále je zaveden tzv. vertikální kontext v obecných gramatikách, který jistým způsobem omezuje možnosti použití jednotlivých kontextových přepisovacích pravidel gramatiky v Kurodově normální formě. Rovněž jsou studovány vlastnosti těchto gramatik s ohledem na zavedená vertikální omezení v průběhu derivačního procesu. V tomto případě je výsledkem radikální snížení vyjadřovací síly gramatik v Kurodově normální formě až na úroveň regulárních jazyků. Jako poslední jsou studovány modifikované bezkontextové gramatiky, které jsou definovány nad volnými grupami místo nad volnými monoidy. Kromě toho byla redukována i množina nonterminálních symbolů, která obsahuje pouze osm nonterminálů. I přes redukci počtu nonterminálních symbolů byla zavedením volných grup síla bezkontextových gramatik zvýšena až na úroveň rekurzívně vyčíslitelných jazyků. Ve všech případech jsou předloženy rigorózní důkazy týkající se vyjadřovacích schopností nově vzniklých struktur.

Klíčová slova

Formální modely, modifikace, vertikální kontext, oboustranné zásobníkové automaty, bezkontextové gramatiky nad volnými grupami, redukce symbolů.

Abstract

The present disertation deals with modifications of various formal models for describing languages. The impact on the generative power of the modified formal models is studied. There are three new formal models investigated. First, the two-sided pushdown automata are defined. Their construction is based on queue grammars. A version with the reduced number of symbols in the pushdown alphabet is also studied. In both cases, by using of two-sided pushdowns, the power of pushdown automata was increased to the level of Turing machines. In the second part, a vertical context in phrase-structure grammars in Kuroda normal form is introduced. This approach limits some applications of rewritting rules in the actual sentential form. The properties of the grammars modified in this way are studied. By the vertical restrictions, the generative power of grammars in Kuroda normal form was remarkably decreased to the level of regular languages. In the last section, there are the modified contex-free grammars presented. These grammars are defined over free groups rather than free monoids. Moreover, the number of nonterminal symbols is reduced to exactly eight nonterminals. Despite it, these grammars generate the family of recursively enumerable languages. In all cases, the rigorous proofs examining the power of the new formal models are presented.

Keywords

Formal models, modifications, vertical context, two-sided pushdown automata, context-free grammars over free groups, symbols reduction.

Obsah

Kapitola 1

Úvod

V současné teorii formálních jazyků existuje obrovské množství různých modelů pro jejich popis a neustále vznikají další. Některé modifikují již známé modely, jiné jsou na těch současných zcela nezávislé. V průběhu uplynulých desetiletí našly některé modely uplatnění zejména v oblasti překladačů, kde představují formální základ pro popis zejména programovacích jazyků, jejich syntaktickou a lexikální analýzu. Velká většina ostatních formálních modelů má pouze teoretický význam. Přestože se z tohoto stavu může na první pohled zdát, že je tvorba a studium dalších variant již zbytečná, není tomu tak. Při studiu vyjadřovacích schopností nových formálních modelů je vždy vhodné najít již známý formální model, na který je možné studovaný formální model transformovat, a u kterého již známe jeho vyjadřovací schopnosti. Pak už zbývá jen dokázat ekvivalenci obou modelů. Čím více formálních modelů bude známých, tím snažší bude studium nových. Z nich pak může některý model vykazovat požadované vlastnosti a tím najít široké uplatnění v praxi.

Pokud při členění formálních modelů pro popis jazyků vezmeme v úvahu historické hledisko, můžeme tyto modely rozdělit i podle toho, jak jsou v oblasti teorie formálních jazyků známé a rozšířené. Získáme tři třídy.

- Základní modely — tato skupina zahrnuje nejznámější modely, které se objevují téměř v každé literatuře a které jsou prověřené i širokým uplatněním v praxi. Jedná se zejména o všechny gramatiky Chomského klasifikace, konečné a zásobníkové automaty a Turingovy stroje.

- Modifikované základní modely — jak již název kategorie napovídá, obsahuje tato skupina základní modely, které byly určitým způsobem modifikovány (ať již rozšířeny nebo omezeny). Jako příklad můžeme jmenovat maticové a programované gramatiky, gramatiky s náhodným kontextem [12], bezkontextové gramatiky nad volnými grupami [2] (všechny tyto modely jsou modifikací bezkontextové gramatiky), vícezásobníkové automaty [11], zásobníkové automaty nad volnými grupami [1], řízené zásobníkové automaty [8] (všechny tyto modely jsou modifikací zásobníkových automatů), atd.

- Ostatní modely — do této skupiny patří všechny ostatní modely, které nelze jednoznačně přiřadit do předchozích dvou skupin (např. z důvodu výrazně odlišného způsobu popisu jazyků). Sem můžeme zařadit např. frontové gramatiky [7].

1.1 Oblasti předkládaného výzkumu

Naším cílem bude studium vybraných základních formálních modelů s ohledem na určité modifikace a omezení, která do těchto modelů zavedeme. Námi vytvořené výsledné modely pak budou spadat do třídy modifikovaných základních modelů.

V této práci budeme studovat následující formální modely:

- Oboustranné zásobníkové automaty — V našem výzkumu provedeme velmi jednoduchou modifikaci zásobníkového automatu. Ta bude spočívat v tom, že do zásobníku umožníme přístup z obou stran. Pokusíme se rovněž zredukovat počet symbolů zásobníkové abecedy. Dále ukážeme, že touto redukcí neztratíme nic z vyjadřovací síly, kterou jsme původní modifikací získali.

- Vertikální kontext v obecných gramatikách — pro tyto modifikace budeme na počátku uvažovat obecnou gramatiku v Kurodově normální formě. Každá gramatika generuje věty jazyka z nějakého výchozího axiomu systematickou aplikací jednotlivých přepisovacích pravidel. Při každé aplikaci je významná pouze aktuální větná forma. Kontext, ve kterém je dané přepisovací pravidlo aplikováno, lze nazvat horizontální.

  My provedeme modifikaci tohoto přístupu v tom smyslu, že budeme uvažovat i kontext vertikální. V konečném důsledku to znamená, že budeme všechny dosud vygenerované větné formy zaznamenávat pod sebe a při aplikaci daného přepisovacího pravidla budeme zkoumat kontext jak aktuální větné formy (horizontální), tak i kontext všech dříve vygenerovaných větných forem (vertikální). S ohledem na tyto skutečnosti zavedeme určitý druh omezení, čímž vytvoříme nový formální model.

- Bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálů — v této práci budeme definovat relaci přímé derivace v bezkontextových gramatikách nad volnými grupami místo nad volnými monoidy a v průběhu derivačního procesu s výhodou využijeme inverzních symbolů a jejich vlastností. Kromě toho omezíme počet nonterminálních symbolů, kdy jich použijeme vždy právě osm.

U čtenáře předpokládáme znalost teorie formálních jazyků a algebry.

Kapitola 2

Oboustranné zásobníkové automaty

Hlavní inspirací pro tento směr výzkumu byly souběžně jednoobrátkové dvouzásobníkové automaty [10]. Jak je v [10] dokázáno, tyto automaty popisují celou třídu rekurzívně vyčíslitelných jazyků.

Naše verze však zavádí nový přístup, zjednodušuje konstrukci pravidel automatu a výrazně zpřehledňuje výsledný důkaz. Přitom rovněž zvyšuje vyjadřovací schopnosti běžných zásobníkových automatů až na úroveň rekurzívně vyčíslitelných jazyků. Navíc je možné pohlížet na takto zavedený zásobník jako na jeden celek. Poznamenejme, že v důkazu jsou využity tzv. frontové gramatiky [7].

Následující pomocná věta bude využita při konstrukci oboustranného zásobníkového automatu.

Lemma 2.1 [10] Pro každý rekurzívně vyčíslitelný jazyk L existuje levě rozšířená frontová gramatika $G = (V, T, W, F, s, P)$ 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 $x \in ((V - T)^* \cup T^*)$.

Podrobný důkaz platnosti této věty pro nás není příliš významný. Popišme si zde alespoň neformálně její význam. Jak lze poznat podle tvarů přepisovacích pravidel, každá levě rozšířená frontová gramatika splňující podmínky této pomocné věty vygeneruje nejprve sekvenci nonterminálních symbolů (první fáze). Jakmile jednou začně generovat terminální symboly (druhá fáze), již nikdy nemůže vygenerovat znovu symboly nonterminální, protože neexistuje žádné pravidlo, které by odebíralo ze začátku fronty některý terminální symbol, a derivace by se tudíž později zablokovala. Jinými slovy, v první fázi jsou generovány pouze nonterminální symboly a ve druhé fázi jsou tyto symboly zpracovávány a generovány symboly terminální. Poslední krok gramatiky odebírá z počátku fronty poslední vygenerovaný nonterminální symbol, generuje na konec fronty poslední sekvenci terminálních symbolů a nastavuje některý z koncových stavových symbolů. Tímto způsobem je generována každá platná věta jazyka, který daná gramatika popisuje. Kromě toho je na samém konci derivace řetězec vygenerovaných nonterminálních symbolů zaznamenán vlevo od symbolu #.

2.1 Základní definice

Uveďme si nyní formální definici oboustranného zásobníkového automatu.

Definice 2.1 — oboustranný zásobníkový automat *Oboustranný zásobníkový automat* je osmice

$$M = (Q, T, Z, R, z, Z_L, Z_R, F_M),$$

kde

- Q je konečná množina stavů
- T je konečná vstupní abeceda
- Z je konečná zásobníková abeceda
- R je konečná množina pravidel tvaru

  $$Z_1|Z_2px \rightarrow \gamma_1|\gamma_2q$$

  kde $Z_1, Z_2 \in Z$, $\gamma_1, \gamma_2 \in Z^*$, $x \in T$ a $p, q \in Q$; pokud $x \in T^*$, nazveme takovýto oboustranný zásobníkový automat *rozšířený*
- z je počáteční stav
- Z_L resp. Z_R je počáteční symbol levé resp. pravé strany zásobníku, $Z_L, Z_R \in Z$
- $F_M \subseteq Q$ je množina koncových stavů

Z algebraického hlediska je množina pravidel R konečná binární relace

$$R \subseteq Z\{|\}ZQT \times Z^*\{|\}Z^*Q$$

$$(\text{případně } R \subseteq Z\{|\}ZQT^* \times Z^*\{|\}Z^*Q),$$

kde každé pravidlo $Z_1|Z_2px \rightarrow \gamma_1|\gamma_2q$ je ve skutečnosti uspořádaná dvojice $(Z_1|Z_2px, \gamma_1|\gamma_2q)$.

Definice 2.2 — konfigurace Nechť

$$M = (Q, T, Z, R, z, Z_L, Z_R, F_M)$$

je oboustranný zásobníkový automat. *Konfigurací* tohoto automatu rozumíme řetězec γqw, kde $q \in Q$, $\gamma \in Z^*$ a $w \in T^*$. Podobně jako u dříve popsaných typů automatů je i zde pomocí konfigurace úplně popsán celkový stav oboustranného zásobníkového automatu v daném okamžiku (γ je aktuální obsah zásobníku, q je aktuální stav a w je aktuální obsah vstupní pásky).

Definice 2.3 — přechod Nechť

$$M = (Q, T, Z, R, z, Z_L, Z_R, F_M)$$

je oboustranný zásobníkový automat. Pokud $L\gamma Rpxy$ a $\gamma_L\gamma\gamma_Rqy$ jsou dvě konfigurace tohoto automatu a $L|Rpx \rightarrow \gamma_L|\gamma_Rq \in R$, pak automat M provádí *přechod* z konfigurace $L\gamma Rpxy$ do konfigurace $\gamma_L\gamma\gamma_Rqy$ podle pravidla $L|Rpx \rightarrow \gamma_L|\gamma_Rq$ a píšeme

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

nebo stručněji $L\gamma Rpxy \vdash \gamma_L\gamma\gamma_Rqy$. Relace $\vdash^n$, $\vdash^+$ a $\vdash^*$ označují posloupnost přechodů délky n pro $n \geq 0$, tranzitivní a reflexivní a tranzitivní uzávěr relace $\vdash$ v tomto pořadí a jsou definovány obvyklým způsobem.

Protože zde definujeme zcela nový model, popišme si zde princip jeho činnosti neformálně. Jak již bylo řečeno v Definici 2.2, představuje každá konfigurace úplný popis celkového stavu automatu v jednom okamžiku. Počáteční konfigurací je s ohledem na definici konfigurace Z_LZ_Rzw, kde první dva symboly představují počáteční symboly levé a pravé strany zásobníku, z je počáteční stav a $w \in T^*$ je přijímaný řetězec, který je na vstupní pásce.

Definice 2.4 — jazyk *Jazyk* přijímaný oboustranným zásobníkovým automatem M je definován jako

$$L(M) = \{w|w \in T^*, Z_L Z_R z w \vdash^* \varepsilon f \varepsilon \text{ a } f \in F_M\}.$$

Z této definice je zřejmé, že 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.

Definice 2.5 — obrátka Nechť M je oboustranný zásobníkový automat a nechť m_1 a m_2 jsou dva po sobě jdoucí přechody prováděné tímto automatem. Jestliže během m_1 automat M nezkrátí zásobník, zatímco během m_2 ano, pak říkáme, že M provádí *obrátku* během m_2. Automat M nazveme *jednoobrátkový*, jestliže provádí nejvýše jednu obrátku během každého výpočetního procesu.

2.2 Vyjadřovací síla

Nyní se již dostáváme k hlavním výsledkům této kapitoly. Označme **2sPDA** (z anglického názvu *two-sided pushdown automata*) třídu jazyků přijímaných oboustrannými zásobníkovými automaty a **2sPDAR** (z angl. *two-sided pushdown automata—reduced*) třídu jazyků přijímaných oboustrannými zásobníkovými automaty s redukovaným počtem symbolů zásobníkové abecedy. Dále označme **RE** třídu rekurzívně vyčíslitelných jazyků.

Věta 2.2 2sPDA = RE

Důkaz

Uvažujme levě rozšířenou frontovou gramatiku

$$G = (V, T, W, F, Sq_0, P)$$

a bez ztráty obecnosti předpokládejme, že G splňuje podmínky popsané v Lemmatu 4.1.

Konstrukce. Nyní budeme konstruovat rozšířený oboustranný zásobníkový automat

$$M = (Q, T, Z, R, z, Z_L, Z_R, F_M)$$

postupnou aplikací následujících kroků.

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

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 $Z_L|Z_R z \to Z_L|SZ_R\langle q_0, 1\rangle$ 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 $Z_L|Z_R\langle q, 1\rangle \to Z_L\overline{A}|xZ_R\langle p, 1\rangle$ 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 $A \in (V - T)$, $q \in (W - F)$, $t \in F$, $y \in T^*$, přidej $Z_L|Z_R\langle q, 2\rangle y \to \overline{A}|\varepsilon f$ do R

6) pro každé $A \in (V - T)$, přidej $\overline{A}|Af \to \varepsilon|\varepsilon f$ do R

Tím 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 režimu *generování nonterminálů*. Podobně jestliže $\langle q, 2\rangle$ je aktuální stav automatu M, říkáme, že M je v režimu *čtení terminálů* (v obou případech pro nějaké $q \in W$).

Nyní je třeba dokázat rovnost $L(G) = L(M)$.

Hlavní myšlenka. M simuluje derivace v levě rozšířené frontové gramatice G s využitím pravidel, které byly zkonstruovány v předcházejícím odstavci. Přitom platí, že automat používá pravidla zkonstruovaná v $b-$tém kroku dříve, než použije pravidla zkonstruovaná v kroku $b + 1$, pro $b = 1, \dots, 5$. Vzhledem k tomu, že levě rozšířená frontová gramatika, která byla použita pro konstrukci, splňuje podmínky popsané v Lemmatu 4.1, objeví se na konci každé úspěšné derivace v této gramatice všechny vygenerované nonterminální symboly vlevo před symbolem # a vpravo zůstane řetězec z T^*F.

Automat začíná vždy v režimu generování nonterminálů a simuluje derivace v G tak, že každý nonterminální symbol (případně řetězec těchto symbolů) vygenerovaný frontovou gramatikou vkládá zprava na zásobník. Při každém derivačním kroku však frontová gramatika zároveň jeden nonterminální symbol bezprostředně vpravo od # načítá a přesouvá jej před # vlevo. Tento symbol (ale s pruhem) vkládá při simulaci automat na zásobník zleva. Na konci simulace každé úspěšné derivace v G je na zásobníku posloupnost symbolů taková, že levá polovina obsahuje posloupnost symbolů s pruhem a pravá polovina posloupnost stejných symbolů bez pruhu v obráceném pořadí. Kromě toho se již automat nachází v některém z koncových stavů. Zbývá splnit druhou podmínku přijetí řetězce, tedy vyprázdnit zásobník. To je zajištěno postupnou aplikací pravidel zkonstruovaných v bodu 6. Vyprázdněním zásobníku je zároveň provedena kontrola korektnosti celé simulace a automat tímto potvrdí přijetí vstupního řetězce. Jakmile se automat není schopen dostat do cílového stavu, případně při vyprazdňování zásobníku nejsou v některém okamžiku oba vrcholy slučitelné, je celý proces zablokován a vstupní řetězec odmítnut. ■

Př.: Uvažujme levě rozšířenou frontovou gramatiku

$$Q = (\{A, B, a, b\}, \{a, b\}, \{p, q, f\}, \{f\}, Ap, \{p_1, p_2, p_3, p_4, p_5, p_6\}),$$

s následujícími pravidly:

$$\begin{aligned} p_1 &: \quad (A, p, AA, p) \\ p_2 &: \quad (A, p, BB, p) \\ p_3 &: \quad (A, p, a, q) \\ p_4 &: \quad (A, q, a, q) \\ p_5 &: \quad (B, q, b, q) \\ p_6 &: \quad (B, q, b, f) \end{aligned}$$

Podle dříve uvedeného postupu zkonstruujeme oboustranný zásobníkový automat

$$M = (Q, T, Z, R, z, Z_L, Z_R, F_M),$$

kde

- $Q = \{f, z, \langle p, 1\rangle, \langle p, 2\rangle, \langle q, 1\rangle, \langle q, 2\rangle\}$
- $T = \{a, b\}$
- $Z = \{Z_L, Z_R, A, B, \overline{A}, \overline{B}\}$
- $F_M = \{f\}$
- *konstrukce množiny pravidel R je uvedena v následující tabulce*

Označení	***Pravidlo***			***Odp. pravidlo z** Q*
1a)	$Z_L \vert Z_R z$	$\rightarrow$	$Z_L \vert A Z_R \langle p, 1\rangle$	—
2a)	$Z_L \vert Z_R \langle p, 1\rangle$	$\rightarrow$	$Z_L \overline{A} \vert AAZ_R \langle p, 1\rangle$	(A, p, AA, p)
2b)	$Z_L \vert Z_R \langle p, 1\rangle$	$\rightarrow$	$Z_L \overline{A} \vert BBZ_R \langle p, 1\rangle$	(A, p, BB, p)
3a)	$Z_L \vert Z_R \langle p, 1\rangle$	$\rightarrow$	$Z_L \vert Z_R \langle p, 2\rangle$	—
3b)	$Z_L \vert Z_R \langle q, 1\rangle$	$\rightarrow$	$Z_L \vert Z_R \langle q, 2\rangle$	—
3c)	$Z_L \vert Z_R \langle f, 1\rangle$	$\rightarrow$	$Z_L \vert Z_R \langle f, 2\rangle$	—
4a)	$Z_L \vert Z_R \langle p, 2\rangle a$	$\rightarrow$	$Z_L \overline{A} \vert Z_R \langle q, 2\rangle$	(A, p, a, q)
4b)	$Z_L \vert Z_R \langle q, 2\rangle a$	$\rightarrow$	$Z_L \overline{A} \vert Z_R \langle q, 2\rangle$	(A, q, a, q)
4c)	$Z_L \vert Z_R \langle q, 2\rangle b$	$\rightarrow$	$Z_L \overline{B} \vert Z_R \langle q, 2\rangle$	(B, q, b, q)
5a)	$Z_L \vert Z_R \langle q, 2\rangle b$	$\rightarrow$	$\overline{B} \vert \varepsilon f$	(B, q, b, f)
6a)	$\overline{A} \vert A f$	$\rightarrow$	$\varepsilon \vert \varepsilon f$	—
6b)	$\overline{B} \vert B f$	$\rightarrow$	$\varepsilon \vert \varepsilon f$	—

*Pro větší přehlednost je ve sloupci **Označení** uveden identifikátor každého pravidla, ve kterém číslo odpovídá číslu kroku v konstrukci a písmeno rozlišuje dané pravidlo v rámci jednoho kroku konstrukce. Sloupec **Pravidlo** obsahuje zkonstruovaná pravidla automatu M a sloupec **Odp. pravidlo z** Q obsahuje korespondující pravidlo z gramatiky Q, ze kterého bylo dané pravidlo M odvozeno.*

Zopakujme, že věta $abb \in L(Q)$ je v levě rozšířené frontové gramatice Q vygenerována derivací

$$
\begin{array}{lll}
 & \#Ap & \\
\Rightarrow & A\#AAp & [(A, p, AA, p)] \\
\Rightarrow & AA\#ABBp & [(A, p, BB, p)] \\
\Rightarrow & AAA\#BBaq & [(A, p, a, q)] \\
\Rightarrow & AAAB\#Babq & [(B, q, b, q)] \\
\Rightarrow & AAABB\#abbf & [(B, q, b, f)].
\end{array}
$$

Nyní si uvedeme posloupnost kroků, kterými přijme stejnou větu oboustranný zásobníkový automat M.

$$
\begin{array}{lll}
 & Z_L Z_R z a b b & \\
\vdash & Z_L A Z_R \langle p,1\rangle abb & 1a \\
\vdash & Z_L \overline{A} AAA Z_R \langle p,1\rangle abb & 2a \\
\vdash & Z_L \overline{AA} AAAABB Z_R \langle p,1\rangle abb & 2b \\
\vdash & Z_L \overline{AA} AAAABB Z_R \langle p,2\rangle abb & 3a \\
\vdash & Z_L \overline{AAA} AAAABB Z_R \langle q,2\rangle bb & 4a \\
\vdash & Z_L \overline{BAAA} AAAABB Z_R \langle q,2\rangle b & 4c \\
\vdash & \overline{BBAAA} AAAABB f \varepsilon & 5a \\
\vdash & \overline{BAAA} AAAAB f \varepsilon & 6b \\
\vdash & \overline{AAA} AAA f \varepsilon & 6b \\
\vdash & \overline{AA} AA f \varepsilon & 6a \\
\vdash & \overline{A} A f \varepsilon & 6a \\
\vdash & \varepsilon f \varepsilon & 6a
\end{array}
$$

Vidíme, že M skončil v konfiguraci $\varepsilon f \varepsilon$. Zásobník je prázdný, vstupní řetězec byl zcela přečten a automat se nachází v koncovém stavu $f \in F_M$. Vstupní řetězec byl tedy úspěšně přijat.

Z výše uvedené konstrukce a důkazu je zřejmé, že zásobníková abeceda obsahuje přibližně dvojnásobný počet symbolů než abeceda $(V - T)$ původní frontové gramatiky. S tím souvisí i počet pravidel zkonstruovaných v bodu 6.

V další větě proto ukážeme, že každý oboustranný zásobníkový automat je možné definovat pouze nad čtyřprvkovou zásobníkovou abecedou. Nazvěme tento automat jako oboustranný zásobníkový automat s redukovaným počtem symbolů zásobníkové abecedy.

Věta 2.3 2sPDAR = RE

Důkaz

Uvažujme znovu levě rozšířenou frontovou gramatiku

$$G = (V, T, W, F, Sq_0, P)$$

a bez ztráty obecnosti předpokládejme, že G splňuje podmínky popsané v Lemmatu 4.1. Konstrukci rozšířeného oboustranného zásobníkového automatu s redukovaným počtem symbolů zásobníkové abecedy

$$M = (Q, T, Z, R, z, 1, 1, F_M)$$

provedeme následujícím způsobem.

Konstrukce. Definujme injektivní zobrazení

$$h : (V - T) \to \{0, 1\}^{n+2}$$

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

$$\overline{h}(A) = \overline{h(A)}^R.$$

Rozšiřme doménu injekce h na $(V-T)^*$. Po tomto rozšíření je h injektivním homomorfismem z $(V-T)^*$ do $(\{0\}\{0,1\}^n\{0\})^*$. Poznamenejme, že k prvkům 0 a 1 existují odpovídající prvky $\overline{0}$ a $\overline{1}$.

Množinu stavů Q, zásobníkovou abecedu Z a množinu koncových stavů F_M vytvoříme následovně.

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

Množina pravidel R je pak vytvořena postupnou aplikací následujících kroků.

1) pro výchozí axiom gramatiky G, Sq_0, 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

2) 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

3) pro každé $q \in W$ přidej $1|1\langle q, 1\rangle \rightarrow 1|1\langle q, 2\rangle$ 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 $1|1\langle q, 2\rangle y \rightarrow 1\overline{h}(A)|1\langle p, 2\rangle$ do R

5) 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

6) přidej do R pravidla $\overline{0}|0f \rightarrow \varepsilon|\varepsilon f$ a $\overline{1}|1f \rightarrow \varepsilon|\varepsilon f$

Tím je konstrukce automatu M kompletní. Analogicky předchozímu důkazu zaveďme následující notaci. Je-li aktuální stav automatu tvaru $\langle q, 1\rangle$, říkáme, že M je v módu *generování nonterminálů*. Podobně je-li aktuální stav automatu tvaru $\langle q, 2\rangle$, říkáme, že M je v módu *čtení terminálů*, $q \in (W-F)$.

Hlavní myšlenka. M simuluje derivace v levě rozšířené frontové gramatice G a kóduje symboly z $(V-T)$ na svém zásobníku pomocí binárních řetězců. V první části předpokládejme, že aktuální větná forma v G je $w\#Avp$, kde $w, v \in (V-T)^*$, $A \in (V-T)$ a $p \in (W-F)$. Odpovídající konfigurace M má potom tvar

$$1\overline{h}(w)h(w)h(A)h(v)1\langle p, 1\rangle\omega,$$

kde $\omega \in T^*$. Nechť $(A, p, x, q) \in P$, kde $x \in (V-T)^*$. Potom,

$$w\#Avp \Rightarrow wA\#vxq$$

v G. V tomto případě musí být M v módu generování nonterminálů a odpovídající pravidlo pro simulaci uvedené derivace automatem M je podle bodu 2 konstrukce

$$1|1\langle p, 1\rangle \rightarrow 1\overline{h}(A)|h(x)1\langle q, 1\rangle \in R.$$

Pomocí tohoto pravidla M přejde do nové konfigurace, která má tvar

$$1\overline{h}(A)\overline{h}(w)h(w)h(A)h(v)h(x)1\langle q, 1\rangle\omega.$$

Všimněme si, že A je zakódován pomocí $\overline{h}$ a výsledný binární řetězec je vložen zleva na zásobník. Dále je pomocí h zakódován řetězec x a výsledek je vložen na zásobník zprava.

Nyní předpokládejme, že $w\#Avup$ je aktuální větná forma v G a $(A, p, y, q) \in P$ je použité pravidlo, kde $u, y \in T^*$, $A \in (V - T)$, $v \in (V - T)*$. Pak

$$w\#Avup \Rightarrow wA\#vuyq$$

v G. Podle bodu 4 konstrukce je odpovídající pravidlo automatu tvaru

$$1|1\langle p, 2\rangle y \rightarrow 1\overline{h}(A)|1\langle q, 2\rangle \in R$$

a M provádí přechod

$$1\overline{h}(w)h(w)h(A)h(v)1\langle p, 2\rangle y\omega' \vdash 1\overline{h}(A)\overline{h}(w)h(w)h(A)h(v)1\langle q, 2\rangle\omega',$$

kde $\omega' \in T^*$. Poznamenejme, že v tomto případě musí být M v módu čtení terminálů. V tomto kroku je zakódován pouze symbol A a výsledný řetězec reprezentovaný pomocí $\overline{h}(A)$ je vložen zleva na zásobník.

Jinými slovy, každé $A \in (V - T)$, které je generováno za symbolem # v G je vloženo jako $h(A)$ zprava na zásobník. Připomeňme, že všechny tyto symboly jsou v každé úspěšné derivaci později v G přesunuty před symbol #. To je důvod, proč M vkládá jejich kódové řetězce tvořené symboly $\overline{0}$ a $\overline{1}$ zleva na zásobník. Na závěr je vyprázdněním zásobníku pomocí pravidel zkonstruovaných v bodu 6 provedena kontrola korektnosti celé simulace. ■

Př.: Příklad je analogický předchozímu příkladu. Jediný rozdíl je v tom, že jsou na zásobníku všechny symboly kódovány binárními řetězci.

Kapitola 3

Vertikální kontext v obecných gramatikách

Běžné gramatiky Chomského hierarchie generují jednotlivé věty jazyka posloupností přímých derivací, kdy je v každém takovémto derivačním kroku aplikováno některé z přepisovacích pravidel na aktuální větnou formu. Tato aplikace není omezena žádnými dalšími podmínkami s výjimkou kontextu (určitého podřetězce aktuální větné formy), který musí být nalezen v aktuální větné formě. Tento kontext můžeme nazývat horizontální. Je zřejmé (a vyplývá to již z definice Chomského gramatik), že nutnou podmínkou pro aplikaci konkrétního přepisovacího pravidla na aktuální větnou formu je výskyt takového podřetězce v této větné formě, který se shoduje s levou stranou aplikovaného přepisovacího pravidla.

3.1 Nový pohled na derivační proces

V této sekci zavedeme zcela nový přístup k derivačnímu procesu. Uvažujme gramatiku

$$G = (V, T, P, S)$$

v Kurodově normální formě a libovolnou derivaci délky n tvaru

$$y_1 \Rightarrow y_2 \Rightarrow \cdots \Rightarrow y_n,$$

kde $y_1 = S$ a $y_n \in L(G)$ pro nějaké $n \geq 1$ v G. Každá takováto derivace může být vyjádřena jako

$$\begin{array}{cccccc} S & \Rightarrow & x_{11} & x_{12} & \dots & x_{1m} \\ & \Rightarrow & x_{21} & x_{22} & \dots & x_{2m} \\ & \vdots & & & & \\ & \Rightarrow & x_{n1} & x_{n2} & \dots & x_{nm}, \end{array}$$

kde $m \geq 1$, $x_{ij} \in V^*$, $x_{i1}x_{i2}\dots x_{im} = y_i$, $1 \leq i \leq n$, $1 \leq j \leq m$.

Protože budeme z tohoto tvaru vycházet v dalším výkladu, definujme si zde proto všechny nezbytné pojmy, které budeme v této souvislosti dále používat.

Každý řetězec $x_{ij} \in V^*$ nazveme *segmentem*. Pokud je navíc daný segment tvořen pouze terminálními symboly, nazveme jej *terminálním segmentem* (např. $x_{n1}, x_{n2}, \dots, x_{nm} \in T^*$). *Sloupcem* nazveme každý vektor segmentů tvaru $\langle x_{1j}, x_{2j}, \dots, x_{nj} \rangle$. Nechť libovolný sloupec tohoto tvaru splňuje $x_{1j} = \varepsilon, x_{2j} = \varepsilon, \dots, x_{k-1j} = \varepsilon$ a $|x_{kj}| = 1, |x_{k+1j}| \geq 1, \dots, |x_{nj}| \geq 1$, kde $1 \leq k < n$ a $x_{kj} \in N$. Potom se x_{kj} nazývá *hlava*. Pomyslný prostor mezi dvěma sousedními sloupci nazveme *hranicí*. Nechť dále xA a By jsou dva sousední segmenty. Pokud existuje v G derivace $xABy \Rightarrow xCDy$ pomocí nějakého pravidla tvaru $AB \to CD \in P$, kde $x, y \in V^*$,

$A, B, C, D \in (V - T)$, říkáme, že G přepisuje $xABy$ (nebo jen A a B) na hranici kontextovým způsobem. Derivaci tohoto druhu nazveme stručně *kontextovou derivací na hranici.*

Nyní zavedeme určitá vertikální omezení do derivačního procesu a budeme zkoumat jejich dopad na generativní schopnosti gramatik v Kurodově normální formě. Vycházejme z dříve zavedených pojmů a zaveďme omezení na maximální počet kontextových derivací na jedné hranici kladnou konečnou konstantou l. Dále předpokládejme, že maximální možná délka každého segmentu je k symbolů. Od této chvíle se tedy budeme dívat na jeden derivační krok v kontextu celé posloupnosti předchozích větných forem.

Protože toto všechno budeme zkoumat v obecné gramatice s využitím Kurodovy normální formy, je zřejmé, že dále zvýšit její sílu již nebude možné. Výsledek našeho výzkumu popisuje následující věta.

Věta 3.1 Jazyk L je regulární, když a jen když existuje konstanta $k \geq 1$ a gramatika

$$G = (V, T, P, S)$$

v Kurodově normální formě taková, že $L = L(G)$ a G generuje každou větu $z \in L(G)$ derivací tvaru

$$\begin{array}{ccllll} S & \Rightarrow & x_{11} & x_{12} & \dots & x_{1m} \\ & \Rightarrow & x_{21} & x_{22} & \dots & x_{2m} \\ & \vdots & & & & \\ & \Rightarrow & x_{n1} & x_{n2} & \dots & x_{nm}, \end{array}$$

kde $n, m \geq 1$, a

1. $|x_{ij}| \leq k$, $1 \leq i \leq n$, $1 \leq j \leq m$
2. pro každé $h = 2, \dots, m$ existuje $x_{rh} \in V^+$ takové, že pro všechna $q = 1, \dots, h$ a $o = h+1, \dots, m$ je $x_{qo} = \varepsilon$
3. pro každé $d = 1, \dots, m-1$ existuje nejvýše l podřetězců tvaru $x_{cd}x_{cd+1}$, kde $1 \leq c \leq n$ takových, že $x_{cd}x_{cd+1}$ je přepsáno na hranici kontextovým způsobem (x_{cd} a x_{cd+1} jsou dva sousední segmenty)

Důkaz

Směr *jen když* je triviální. Zbývá proto dokázat směr *když.* Uvažujme tedy libovolnou gramatiku

$$G = (V, T, P, S)$$

v Kurodově normální formě, která splňuje podmínky dané výše uvedenou větou a definujme $P = \{AB \rightarrow CD | AB \rightarrow CD \in P\}$.

Konstrukce. Zkonstruujme rozšířený konečný automat

$$M = (Q, T, R, s, \{f\})$$

aplikací následujících pravidel.

- $Q = \{f\} \cup \{\langle A, u, \alpha, v\rangle | \langle A, u, \alpha, v\rangle \neq f,\ A \in N,\ \alpha \in V^*,\ |\alpha| \leq k,\ u, v \in P_{CS}^*,\ |u|, |v| \leq l\}$
- $s = \langle S, \varepsilon, \varepsilon, \varepsilon\rangle$

- Množina $R = R_{IN} \cup R_{CS1} \cup R_{CS2} \cup R_I \cup R_{HEAD} \cup R_F$ je zkonstruována následovně:

  I pro každé $x \in T^*$, kde $|x| \leq k$, přidej do R_I pravidla tvaru $\langle X, u, \varepsilon, \varepsilon \rangle x \rightarrow \langle X, u, x, \varepsilon \rangle$

  II pro každé $a \in T \cup \{\varepsilon\}$ a pravidlo $A \rightarrow a \in P$, kde $1 \leq |\alpha a \beta| \leq k$, přidej do R_{IN} pravidla tvaru $\langle X, u, \alpha a \beta, v \rangle \rightarrow \langle X, u, \alpha A \beta, v \rangle$

  III pro každé $AB \rightarrow CD \in P$ přidej:

  (a) $\langle X, u, \alpha C, v \rangle \rightarrow \langle X, u, \alpha A, vAB \rightarrow CD \rangle$ do R_{CS1}

  (b) $\langle X, AB \rightarrow CDu, D\beta, v \rangle \rightarrow \langle X, u, B\beta, v \rangle$ do R_{CS2}

  (c) $\langle X, u, \alpha CD\beta, v \rangle \rightarrow \langle X, u, \alpha AB\beta, v \rangle$ do R_{IN}

  IV pro každé $A \rightarrow BC \in P$ přidej:

  (a) $\langle X, u, \alpha BC\beta, v \rangle \rightarrow \langle X, u, \alpha A\beta, v \rangle$ do R_{IN}

  (b) $\langle A, \varepsilon, B, v \rangle \rightarrow \langle C, v, \varepsilon, \varepsilon \rangle$ do R_{HEAD}

  V pro každé $A \in N$, přidej $\langle A, \varepsilon, A, \varepsilon \rangle \rightarrow f$ do R_F

Tím je konstrukce hotova. Nyní je třeba dokázat rovnost obou jazyků $L(G) = L(M)$.

Hlavní myšlenka. Popišme neformálně jednotlivé komponenty stavů automatu.

1. První komponenta představuje hlavu aktuálního sloupce.
2. Ve druhé komponentě je zaznamenávána posloupnost kontextových pravidel tvaru
$$AB \rightarrow CD,$$
která byla aplikována na pravé straně hranice v předchozím sloupci.
3. Ve třetí komponentě je zaznamenán aktuální segment a simulovány redukce jak uvnitř sloupce, tak na jeho hranicích.
4. Ve čtvrté komponentě jsou uložena postupně všechna kontextová pravidla, která byla aplikována na pravé hranici aktuálního sloupce. Pomocí pravidel zkonstruovaných v bodu IVb je tento obsah při vytváření nové hlavy přesunut do druhé komponenty.

Automat M postupně načítá řetězce o maximální délce k symbolů (terminální segmenty). Po načtení každého terminálního segmentu simuluje jak redukce uvnitř sloupce, tak i redukce na hranicích, kde mohla být aplikována pravidla z G kontextovým způsobem. Přitom jsou redukce na pravé hranici sloupce zaznamenávány ve čtvrté komponentě stavu automatu. Před načtením dalšího terminálního segmentu je zaznamenaný řetězec přesunut do druhé komponenty stavu a v následujícím sloupci je provedena kontrola, při jejímž úspěšném dokončení dojde k vyprázdnění druhé komponenty stavu automatu. Po úspěšné redukci aktuálního sloupce je ustavena nová hlava a načten další terminální segment. Pokud je vstupní řetězec zcela přečten, jsou provedeny všechny zbylé redukce a první a třetí komponenta stavu automatu jsou si rovny, M přechází do koncového stavu a vstupní řetězec je přijat. ■

Př.: Stejně jako v předchozích důkazech si uvedeme i zde praktický příklad, na kterém bude vidět princip činnosti konečného automatu přijímajícího jazyk definovaný gramatikou s vertikálními omezeními danými větou 3.1. Uvažujme tedy gramatiku v Kurodově normální formě

$$G = (V, T, P, A),$$

kde:

- $V = \{A, B, C, D, E, G, H, P, Q, b, g, p, q\}$
- $T = \{b, g, p, q\}$
- $P = \{$
  $$\begin{array}{lcl} A & \rightarrow & BC, \\ C & \rightarrow & DE, \\ DE & \rightarrow & GH, \\ H & \rightarrow & HG, \\ HG & \rightarrow & PQ, \\ B & \rightarrow & b, \\ G & \rightarrow & g, \\ P & \rightarrow & p, \\ Q & \rightarrow & q \end{array}$$
  $\}$
- *startovací symbol gramatiky je* $A \in (V - T)$

Dále uvažujme následující derivaci v této gramatice, která generuje větu bgpq.

$$\begin{array}{lll} & A & \\ \Rightarrow & BC & [A \rightarrow BC] \\ \Rightarrow & BDE & [C \rightarrow DE] \\ \Rightarrow & BGH & [DE \rightarrow GH] \\ \Rightarrow & BGHG & [H \rightarrow HG] \\ \Rightarrow & bGHG & [B \rightarrow b] \\ \Rightarrow & bgHG & [G \rightarrow g] \\ \Rightarrow & bgPQ & [HG \rightarrow PQ] \\ \Rightarrow & bgpQ & [P \rightarrow p] \\ \Rightarrow & bgpq & [Q \rightarrow q] \end{array}$$

Omezme maximální šířku sloupce na $k = 2$ *a maximální počet kontextových derivací na hranici na* $l = 1$ *a zkonstruujme konečný automat*

$$M = (Q, T, R, \langle A, \varepsilon, \varepsilon, \varepsilon \rangle, \{f\}),$$

který simuluje derivace v gramatice G. Vzhledem k vysoké složitosti konstrukce takovéhoto konečného automatu a vzhledem k neúměrně velkému počtu přechodových pravidel a stavů, které jsou konstrukcí generovány, uvedeme pouze nezbytně nutné stavy a přechodová pravidla pro náš příklad. U každého přechodového pravidla navíc uvedeme číslo bodu konstrukce, podle kterého bylo vytvořeno, a odpovídající přepisovací pravidlo z G.

- $Q = \{$
  $$\begin{array}{lll} \langle A, \varepsilon, \varepsilon, \varepsilon \rangle, & \langle A, \varepsilon, b, \varepsilon \rangle, & \langle A, \varepsilon, B, \varepsilon \rangle, \\ \langle C, \varepsilon, \varepsilon, \varepsilon \rangle, & \langle C, \varepsilon, g, \varepsilon \rangle, & \langle C, \varepsilon, G, \varepsilon \rangle, \\ \langle C, \varepsilon, D, DE \rightarrow GH \rangle, & \langle E, DE \rightarrow GH, \varepsilon, \varepsilon \rangle, & \langle E, DE \rightarrow GH, pq, \varepsilon \rangle, \\ \langle E, DE \rightarrow GH, pQ, \varepsilon \rangle, & \langle E, DE \rightarrow GH, PQ, \varepsilon \rangle, & \langle E, DE \rightarrow GH, HG, \varepsilon \rangle, \\ \langle E, DE \rightarrow GH, H, \varepsilon \rangle, & \langle E, \varepsilon, E, \varepsilon \rangle, & \text{f}, \ldots \end{array}$$
  $\}$
- *R obsahuje (mimo dalších zde neuvedených) tato pravidla:*

Označení	***Pravidlo***			***Odp. prav. z*** G
I-1	$\langle A, \varepsilon, \varepsilon, \varepsilon \rangle b$	$\rightarrow$	$\langle A, \varepsilon, b, \varepsilon \rangle$	—
II-1	$\langle A, \varepsilon, b, \varepsilon \rangle$	$\rightarrow$	$\langle A, \varepsilon, B, \varepsilon \rangle$	$B \rightarrow b$
IVb-1	$\langle A, \varepsilon, B, \varepsilon \rangle$	$\rightarrow$	$\langle C, \varepsilon, \varepsilon, \varepsilon \rangle$	$A \rightarrow BC$
I-2	$\langle C, \varepsilon, \varepsilon, \varepsilon \rangle g$	$\rightarrow$	$\langle C, \varepsilon, g, \varepsilon \rangle$	—
II-2	$\langle C, \varepsilon, g, \varepsilon \rangle$	$\rightarrow$	$\langle C, \varepsilon, G, \varepsilon \rangle$	$G \rightarrow g$
IIIa	$\langle C, \varepsilon, G, \varepsilon \rangle$	$\rightarrow$	$\langle C, \varepsilon, D, DE \rightarrow GH \rangle$	$DE \rightarrow GH$
IVb-2	$\langle C, \varepsilon, D, DE \rightarrow GH \rangle$	$\rightarrow$	$\langle E, DE \rightarrow GH, \varepsilon, \varepsilon \rangle$	$C \rightarrow DE$
I-3	$\langle E, DE \rightarrow GH, \varepsilon, \varepsilon \rangle pq$	$\rightarrow$	$\langle E, DE \rightarrow GH, pq, \varepsilon \rangle$	—
II-3	$\langle E, DE \rightarrow GH, pq, \varepsilon \rangle$	$\rightarrow$	$\langle E, DE \rightarrow GH, pQ, \varepsilon \rangle$	$Q \rightarrow q$
II-4	$\langle E, DE \rightarrow GH, pQ, \varepsilon \rangle$	$\rightarrow$	$\langle E, DE \rightarrow GH, PQ, \varepsilon \rangle$	$P \rightarrow p$
IIIc	$\langle E, DE \rightarrow GH, PQ, \varepsilon \rangle$	$\rightarrow$	$\langle E, DE \rightarrow GH, HG, \varepsilon \rangle$	$HG \rightarrow PQ$
IVa	$\langle E, DE \rightarrow GH, HG, \varepsilon \rangle$	$\rightarrow$	$\langle E, DE \rightarrow GH, H, \varepsilon \rangle$	$H \rightarrow HG$
IIIb	$\langle E, DE \rightarrow GH, H, \varepsilon \rangle$	$\rightarrow$	$\langle E, \varepsilon, E, \varepsilon \rangle$	$DE \rightarrow GH$
V	$\langle E, \varepsilon, E, \varepsilon \rangle$	$\rightarrow$	f	—

Označování jednotlivých přechodových pravidel je provedeno podle následující konvence. Římská část popř. římská část společně s písmenem označuje konkrétní bod konstrukce, ve kterém jsou pravidla tohoto tvaru konstruována. Arabská číslice (pokud je uvedena) rozlišuje více pravidel zkonstruovaných v tomtéž bodu konstrukce.

Nyní máme zkonstruovaný konečný automat a můžeme si tedy uvést posloupnost přechodů, kterou je přijat řetězec bgpq. Jednotlivé kroky stručně okomentujeme.

	$\langle A, \varepsilon, \varepsilon, \varepsilon \rangle bgpq$		
$\vdash$	$\langle A, \varepsilon, b, \varepsilon \rangle gpq$	I-1	*načtení b ze vstupní pásky (1. term. segment)*
$\vdash$	$\langle A, \varepsilon, B, \varepsilon \rangle gpq$	II-1	*redukce podle* $B \rightarrow b$
$\vdash$	$\langle C, \varepsilon, \varepsilon, \varepsilon \rangle gpq$	IVb-1	*ustavení nové hlavy podle* $A \rightarrow BC$
$\vdash$	$\langle C, \varepsilon, g, \varepsilon \rangle pq$	I-2	*načtení g ze vstupní pásky*
$\vdash$	$\langle C, \varepsilon, G, \varepsilon \rangle pq$	II-2	*redukce podle* $G \rightarrow g$
$\vdash$	$\langle C, \varepsilon, D, DE \rightarrow GH \rangle pq$	IIIa	*viz Komentář 1 níže*
$\vdash$	$\langle E, DE \rightarrow GH, \varepsilon, \varepsilon \rangle pq$	IVb-2	*ustavení nové hlavy podle* $C \rightarrow DE$
$\vdash$	$\langle E, DE \rightarrow GH, pq, \varepsilon \rangle$	I-3	*načtení pq ze vstupní pásky*
$\vdash$	$\langle E, DE \rightarrow GH, pQ, \varepsilon \rangle$	II-3	*redukce podle* $Q \rightarrow q$
$\vdash$	$\langle E, DE \rightarrow GH, PQ, \varepsilon \rangle$	II-4	*redukce podle* $P \rightarrow p$
$\vdash$	$\langle E, DE \rightarrow GH, HG, \varepsilon \rangle$	IIIc	*redukce podle* $HG \rightarrow PQ$ *uvnitř sloupce (!)*
$\vdash$	$\langle E, DE \rightarrow GH, H, \varepsilon \rangle$	IVa	*redukce podle* $H \rightarrow HG$ *uvnitř sloupce (!)*
$\vdash$	$\langle E, \varepsilon, E, \varepsilon \rangle$	IIIb	*viz Komentář 2 níže*
$\vdash$	f	V	*viz Komentář 3 níže*

Komentář 1: simulace první části aplikace kontextového pravidla $DE \rightarrow GH$ *(redukce* G *na* D*) a uložení tohoto pravidla do čtvrté komponenty stavu automatu*

Komentář 2: simulace druhé části aplikace kontextového pravidla $DE \rightarrow GH$ *(redukce* H *na* E *a zároveň i kontrola odebráním tohoto pravidla z počátku řetězce ve druhé komponentě stavu)*

Komentář 3: přechod do koncového stavu a úspěšný konec simulace (přijetí řetězce)

V uvedeném příkladu byla simulována derivace z gramatiky G*, která byla tvořena třemi sloupci. První resp. druhý sloupec vznikl načtením terminálního segmentu b resp. g. Třetí sloupec vznikl* *načtením posledního terminálního segmentu pq. Zde je potřeba zdůraznit, že je celý proces nedeterministický. Pokud by v některé z konfigurací nebylo aplikovatelné žádné přechodové pravidlo, celá simulace by byla zablokována a vstupní řetězec by nebyl přijat. Přijímání jazyka generovaného gramatikou G by pak bylo zajištěno jiným automatem, s jiným počtem a šířemi sloupců, případně horní hranicí počtu aplikací kontextových pravidel na hranici. Jedinou podmínkou je, že maximální šíře sloupců a maximální počet aplikací kontextových pravidel na hranici musí být konečné.*

Kapitola 4

Bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálů

V této kapitole budeme studovat další nový model pro popis formálních jazyků, který je založený na bezkontextových gramatikách. Ukážeme, že velmi jednoduchou a přirozenou modifikací bezkontextové gramatiky lze výrazně zvýšit její vyjadřovací schopnosti. Jak již název kapitoly napovídá, tyto gramatiky mají rovněž omezen počet nonterminálních symbolů—přesněji řečeno, bude jich právě osm.

Poznamenejme, že bezkontextové gramatiky nad volnými grupami jsou rovněž studovány v [3] a v [4], kde je popisována jejich neredukovaná varianta. Kromě toho je však ukázáno, že podobný přístup je aplikovatelný i v případě E0L gramatik, které pracují na rozdíl od běžných bezkontextových gramatik paralelně.

Již v kapitole 3.1 jsme definovali gramatiky Chomského hierarchie a relaci přímé derivace v těchto gramatikách. Přestože se bezkontextová gramatika nad volnou grupou téměř neliší od běžné bezkontextové gramatiky, uvedeme si zde její úplnou definici včetně definic dalších souvisejících pojmů.

4.1 Základní definice

Definice 4.1 — bezkontextová gramatika nad volnou grupou s redukovaným počtem nonterminálů *Bezkontextová gramatika nad volnou grupou s redukovaným počtem nonterminálů* (dále jen $\mathbf{CF}^{\circ}\mathbf{R}$ gramatika podle anglického označení *context-free—reduced*, kde $^{\circ}$ vyjadřuje skutečnost, že je gramatika definována nad volnou grupou) je čtveřice

$$G = (V, T, P, S),$$

kde

- V je konečná abeceda nazývaná jako úplná abeceda gramatiky G
- $T \subset V$ je abeceda terminálních symbolů
- $N = V - T = \{S, \overline{S}, 0, \overline{0}, 1, \overline{1}, 2, \overline{2}\}$ je abeceda nonterminálních symbolů
- P je konečná množina přepisovacích pravidel tvaru

$$A \to \alpha,$$

kde $A \in N$ a $\alpha \in V^\circ$ (připomeňme, že V° je volná grupa generovaná abecedou V a operací konkatenace)

- $S \in N$ je startovací symbol

Protože tato gramatika definuje věty jazyka systematickou aplikací jednotlivých přepisovacích pravidel, definujme si zde její relaci přímé derivace a derivace.

Definice 4.2 — přímá derivace Nechť $G = (V, T, P, S)$ je **CF°R** gramatika a nechť $\lambda, \mu \in V^\circ$ jsou řetězce takové, že

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

a

$$\mu = \alpha\gamma\beta.$$

Jestliže $A \rightarrow \gamma \in P$, pak mezi řetězci λ a μ platí relace $\circ\Rightarrow$ nazvaná *přímá derivace*, píšeme

$$\lambda \circ\Rightarrow \mu[A \rightarrow \gamma]$$

nebo stručněji $\lambda \circ\Rightarrow \mu$ a říkáme, že μ lze přímo derivovat z λ v G.

Definice 4.3 — derivace Nechť $G = (V, T, P, S)$ je **CF°R** gramatika.

1. pro každé $u \in V^\circ$ platí $u \circ\Rightarrow^0 u[\varepsilon]$ (identita)
2. nechť $u_0, \ldots, u_n \in V^\circ$, pro $n \geq 1$ a $u_{i-1} \circ\Rightarrow u_i[p_i]$, kde $p_i \in P$ pro $i = 1, 2, \ldots, n$, tedy

   $$u_0 \circ\Rightarrow u_1[p_1] \circ\Rightarrow u_2[p_2] \circ\Rightarrow \ldots \circ\Rightarrow u_n[p_n];$$

   tuto posloupnost přímých derivací v G nazýváme *derivací délky* n a píšeme

   $$u_0 \circ\Rightarrow^n u_n[p_1 p_2 \ldots p_n]$$

   nebo stručněji $u_0 \circ\Rightarrow^n u_n$

Relace $\circ\Rightarrow^n$ představuje $n-$tou mocninu relace $\circ\Rightarrow$. Za tohoto předpokladu potom definujeme relace $\circ\Rightarrow^+$, pokud $n \geq 1$ a $\circ\Rightarrow^*$, pokud $n \geq 0$, které reprezentují tranzitivní uzávěr a reflexivní a tranzitivní uzávěr relace $\circ\Rightarrow$ v tomto pořadí.

Definice 4.4 — větná forma, věta Nechť $G = (V, T, P, S)$ je **CF°R** gramatika. Platí-li v G derivace

$$S \circ\Rightarrow^* \alpha$$

pro nějaké $\alpha \in V^\circ$, nazývá se α *větnou formou.* Pokud $\alpha \in T^\circ$, nazývá se *věta.*

Definice 4.5 — jazyk generovaný CF°R gramatikou Nechť $G = (V, T, P, S)$ je **CF°R** gramatika. *Jazyk* $L(G)^\circ$ generovaný touto gramatikou je definován jako

$$L(G)^\circ = \{w \in T^\circ | S \circ\Rightarrow^* w\}.$$

Srovnáme-li tyto definice s definicemi 3.1 a 3.2 z kapitoly 3.1.1 a s definicemi z kapitoly 3.1.2, najdeme pouze dvě odlišnosti od běžných bezkontextových gramatik. Tou první a zcela zásadní je skutečnost, že jsou veškeré derivace v **CF°R** gramatice definovány nad volnou grupou V° generovanou abecedou V místo nad volným monoidem V^*. Druhá odlišnost je pouze formálního rázu. Abychom zdůraznili, že je derivace prováděna nad volnou grupou, značíme relaci přímé derivace, derivaci délky n, tranzitivní uzávěr ralace přímé derivace a reflexivní a tranzitivní uzávěr relace přímé derivace postupně symboly $\circ\Rightarrow$, $\circ\Rightarrow^n$, $\circ\Rightarrow^+$ a $\circ\Rightarrow^*$.

4.2 Vyjadřovací síla

Ještě než přejdeme k hlavnímu důkazu demonstrujícímu vyjadřovací schopnosti **CF°R** gramatik, je nutné uvést následující pomocnou větu, na níž se budeme v dalším důkazu odvolávat.

Lemma 4.1 Pro každou gramatiku $H = (V, T, P, S)$ typu 0 existuje ekvivalentní gramatika $G = (V_G, T, P_G, S)$ typu 0 taková, že každé pravidlo v P_G má jeden z následujících tvarů:

1. $AB \to CD$, kde $A \neq C$
2. $A \to BC$, kde $A \neq B$
3. $A \to x$

Přitom $V_G = N_G \cup T$, $A, B, C, D \in N_G$ a $x \in T \cup \{\varepsilon\}$.

Důkaz

Nechť $H = (V, T, P, S)$ je gramatika, $N = V - T$. Bez ztráty obecnosti předpokládejme, že H je v Kurodově normální formě. Definujme gramatiku $G = (V_G, T, P_G, S)$, $V_G = N_G \cup T$, kde N_G a P_G jsou zkonstruovány následovně:

a) přiřaď $N_G = N$ a přidej všechna pravidla z P, která splňují body 1 až 3, do P_G

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

c) 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

Formální důkaz, že H a G jsou ekvivalentní je jednoduchý a ponecháme jej na čtenáři. ■

Nyní již můžeme uvést větu týkající se vyjadřovacích schopností **CF°R** gramatik, jejíž důkaz zároveň představuje hlavní výsledek této kapitoly.

Věta 4.2 CF°R=RE

Důkaz

Nechť $G = (V, T, P, S)$ je gramatika typu 0, $N = V - T$. Bez ztráty obecnosti předpokládejme, že G splňuje Lemmat 1 a Je tedy zároveň i v Kurodově normální formě.

Konstrukce. Zkonstruujme **CF°R** gramatiku

$$\Gamma = (V_\Gamma, T, P_\Gamma, S_\Gamma),$$

kde

$$N_\Gamma = V_\Gamma - T = \{S_\Gamma, \overline{S_\Gamma}, 0, \overline{0}, 1, \overline{1}, 2, \overline{2}\}.$$

Pro další důkaz tiše předpokládáme, že abeceda terminálů T gramatiky Γ obsahuje ke každému terminálnímu symbolu i jeho inverzní variantu, která je vynucena volnou grupou. Přesto ji budeme dále značit symbolem T. Čímž zároveň zdůrazníme, že jazyk generovaný gramatikou Γ je definován nad symboly stejné abecedy jako je tomu u gramatiky G.

Definujme nyní injektivní zobrazení

$$g : N \to \{0, 1\}^n$$

a

$$h : N \to \{0, 1\}^{2n}$$

taková, že

$$h(A) = g(A)rev(g(A)),$$

kde $A \in N$ a

$$n = \lceil \log_2(card(N)) \rceil.$$

Poznamenejme, že inverzní symboly k $0, 1, 2 \in N_\Gamma$ jsou $\overline{0}, \overline{1}, \overline{2} \in N_\Gamma$ v tomto pořadí. Jestliže

$$h(A) = a_1 \ldots a_n a_n \ldots a_1,$$

$a_i \in \{0, 1\}$ pro $i = 1, 2, \ldots, n$, $n \geq 1$, pak

$$\overline{h}(A) = \overline{a_1} \ldots \overline{a_n a_n} \ldots \overline{a_1}.$$

Množina přepisovacích pravidel P_Γ je zkonstruována následovně:

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

Zde platí, že $A, B, C, D \in N$ a $x \in T \cup \{\varepsilon\}$. Konstrukce gramatiky Γ je nyní kompletní.

Hlavní myšlenka. Γ binárním způsobem kóduje nonterminály z G a simuluje aplikace bezkontextových pravidel tvaru $A \to BC$ z G (viz III) tak, že přepisuje symbol 2, který následuje $h(A)$, na $\overline{h}(A)\overline{2}2h(B)2h(C)2$. Výsledkem tohoto kroku je podřetězec $h(A)\overline{h}(A)$ vygenerovaný v nové větné formě. Tento (korektní) tvar je vymazán redukcí volné grupy. Jakýkoliv jiný tvar—to jest odlišný od tvaru $h(A)\overline{h}(A)$—nemůže být tímto způsobem eliminován, takže není možné vygenerovat platnou větu jazyka. Γ simuluje analogicky aplikace pravidel tvaru $A \to x$ z bodu IV. Podobným způsobem jsou v Γ simulovány i aplikace pravidel tvaru $AB \to CD$ (viz II). V tomto případě je však přepsán symbol 2, který následuje podřetězec $h(A)2h(B)$, na řetězec začínající inverzním binárním kódem obou přepisovaných nonterminálů $\overline{h}(B)\overline{2}\overline{h}(A)$. ■

Př.: Stejně jako v předchozích důkazech si i zde uvedeme před rigorózním důkazem konkrétní příklad. Jako výchozí gramatiku pro konstrukci ***CF°R*** *gramatiky uvažujme gramatiku*

$$G = (V, T, P, S)$$

v Kurodově normální formě, kde

- $V = \{B, C, G, P, Q, R, S, X, Y, Z, x, y, z\}$
- $T = \{x, y, z\}$
- $P = \{$

$$
\begin{array}{llll}
S \to XP, & P \to YZ, & S \to XQ, & Q \to GR, \\
R \to BZ, & GB \to SB, & ZB \to CZ, & CZ \to BZ, \\
YB \to YY, & X \to x, & Y \to y, & Z \to z
\end{array}
$$

}

- *startovací symbol gramatiky je S*

Tato gramatika je již v Kurodově normální formě a generuje známý jazyk $x^i y^i z^i$, *kde* $i \geq 1$, *který není bezkontextový. Snadno ověříme, že G rovněž splňuje podmínky dané Lemmatem 4.1.*

Nyní zkonstruujeme ***CF°R*** *gramatiku*

$$\Gamma = (V_\Gamma, T, P_\Gamma, S_\Gamma).$$

Úplná abeceda této gramatiky je tvořena množinou

$$V_\Gamma = \{S_\Gamma, \overline{S_\Gamma}, 0, \overline{0}, 1, \overline{1}, 2, \overline{2}, x, y, z, \overline{x}, \overline{y}, \overline{z}\}.$$

Množina nonterminálních symbolů gramatiky Γ *je pak definována jako*

$$N_\Gamma = V_\Gamma - T = \{S_\Gamma, \overline{S_\Gamma}, 0, \overline{0}, 1, \overline{1}, 2, \overline{2}\}.$$

Délku binárních kódových slov pro jednotlivé nonterminální symboly původní gramatiky vypočítáme podle vztahu

$$n = \lceil \log_2(card(V - T)) \rceil = \lceil \log_2(10) \rceil = 4$$

Zobrazení $g : N \to \{0, 1\}$ *definujme v následující tabulce.*

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
g(B) & = & 0001 & g(C) & = & 0010 \\
g(G) & = & 0011 & g(P) & = & 0100 \\
g(Q) & = & 0101 & g(R) & = & 0110 \\
g(S) & = & 0111 & g(X) & = & 1000 \\
g(Y) & = & 1001 & g(Z) & = & 1010
\end{array}
$$

Zobrazení $h : N \to \{0, 1\}^{2n}$ *a* $\overline{h} : N \to \{\overline{0}, \overline{1}\}^{2n}$ *již definují samotná kódová slova pro jednotlivé nonterminální symboly z gramatiky G a jejich inverzní varianty a jsou uvedena v další tabulce.*

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
h(B) & = & 00011000 & \overline{h}(B) & = & \overline{00011000} \\
h(C) & = & 00100100 & \overline{h}(C) & = & \overline{00100100} \\
h(G) & = & 00111100 & \overline{h}(G) & = & \overline{00111100} \\
h(P) & = & 01000010 & \overline{h}(P) & = & \overline{01000010} \\
h(Q) & = & 01011010 & \overline{h}(Q) & = & \overline{01011010} \\
h(R) & = & 01100110 & \overline{h}(R) & = & \overline{01100110} \\
h(S) & = & 01111110 & \overline{h}(S) & = & \overline{01111110} \\
h(X) & = & 10000001 & \overline{h}(X) & = & \overline{10000001} \\
h(Y) & = & 10011001 & \overline{h}(Y) & = & \overline{10011001} \\
h(Z) & = & 10100101 & \overline{h}(Z) & = & \overline{10100101}
\end{array}
$$

V tuto chvíli máme již připraveno vše potřebné k tomu, abychom mohli zkonstruovat jednotlivá přepisovací pravidla gramatiky Γ. *Jejich konstrukce je přehledně uvedena v následující tabulce. Význam hodnot ve sloupci* ***Označení*** *je takový, že římská část udává číslo kroku konstrukce a písmeno rozlišuje dané pravidlo v rámci jednoho bodu konstrukce. Ve sloupci* ***Odp. pravidlo z G*** *je uvedeno původní přepisovací pravidlo z gramatiky G, podle kterého bylo dané bezkontextové pravidlo pro gramatiku* Γ *zkonstruováno. Pro větší přehlednost budeme v jednotlivých zkonstruovaných přepisovacích pravidlech uvádět oddělovací nonterminál* ***2*** *tučným písmem.*

Označení			***Pravidlo***	***Odp. pravidlo z*** G
Ia	S_Γ	$\rightarrow$	$01111110\mathbf{2}$	—
IIa	2	$\rightarrow$	$\overline{000110002001111002}\mathbf{2}01111110\mathbf{2}00011000\mathbf{2}$	$GB \rightarrow SB$
IIb	2	$\rightarrow$	$\overline{000110002101001012}\mathbf{2}00100100\mathbf{2}10100101\mathbf{2}$	$ZB \rightarrow CZ$
IIc	2	$\rightarrow$	$\overline{101001012001001002}\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	$CZ \rightarrow BZ$
IId	2	$\rightarrow$	$\overline{000110002100110012}\mathbf{2}10011001\mathbf{2}10011001\mathbf{2}$	$YB \rightarrow YY$
IIIa	2	$\rightarrow$	$\overline{011111102}\mathbf{2}10000001\mathbf{2}01000010\mathbf{2}$	$S \rightarrow XP$
IIIb	2	$\rightarrow$	$\overline{010000102}\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}$	$P \rightarrow YZ$
IIIc	2	$\rightarrow$	$\overline{011111102}\mathbf{2}10000001\mathbf{2}01011010\mathbf{2}$	$S \rightarrow XQ$
IIId	2	$\rightarrow$	$\overline{010110102}\mathbf{2}00111100\mathbf{2}01100110\mathbf{2}$	$Q \rightarrow GR$
IIIe	2	$\rightarrow$	$\overline{011001102}\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	$R \rightarrow BZ$
IVa	2	$\rightarrow$	$\overline{10000001}x$	$X \rightarrow x$
IVb	2	$\rightarrow$	$\overline{10011001}y$	$Y \rightarrow y$
IVc	2	$\rightarrow$	$\overline{10100101}z$	$Z \rightarrow z$

Vynegerujme nyní větu xxyyzz. Následující posloupnost představuje derivaci této věty v původní gramatice G.

$$
\begin{array}{lll}
 & \mathrm{S} & \\
\Rightarrow & XQ & [S \rightarrow XQ] \\
\Rightarrow & XGR & [Q \rightarrow GR] \\
\Rightarrow & XGBZ & [R \rightarrow BZ] \\
\Rightarrow & XSBZ & [GB \rightarrow SB] \\
\Rightarrow & XXPBZ & [S \rightarrow XP] \\
\Rightarrow & XXYZBZ & [P \rightarrow YZ] \\
\Rightarrow & XXYCZZ & [ZB \rightarrow CZ] \\
\Rightarrow & XXYBZZ & [CZ \rightarrow BZ] \\
\Rightarrow & XXYYZZ & [YB \rightarrow YY] \\
\Rightarrow & xXYYZZ & [X \rightarrow x] \\
\Rightarrow & xxYYZZ & [X \rightarrow x] \\
\Rightarrow & xxyYZZ & [Y \rightarrow y] \\
\Rightarrow & xxyyZZ & [Y \rightarrow y] \\
\Rightarrow & xxyyzZ & [Z \rightarrow z] \\
\Rightarrow & xxyyzz & [Z \rightarrow z]
\end{array}
$$

Nyní vygenerujeme stejnou větu dříve zkonstruovanou ***CF°R*** *gramatikou* Γ. *Vpravo od každé větné formy uvedeme buď označení přepisovacího pravidla, pomocí kterého byla daná větná forma vygenerována, nebo "redukce", pokud byla větná forma získána z předchozí větné formy redukcí inverzních podřetězců s využitím vlastnosti volné grupy.*

	S_Γ	
$\Rightarrow$	$01111110\mathbf{2}$	Ia
$\Rightarrow$	$01111110\overline{011111102}\mathbf{2}10000001\mathbf{2}01011010\mathbf{2}$	IIIc
$=$	$10000001\mathbf{2}01011010\mathbf{2}$	redukce
$\Rightarrow$	$10000001\mathbf{2}01011010\overline{010110102}\mathbf{2}00111100\mathbf{2}01100110\mathbf{2}$	IIId
$=$	$10000001\mathbf{2}00111100\mathbf{2}01100110\mathbf{2}$	redukce
$\Rightarrow$	$10000001\mathbf{2}00111100\mathbf{2}01100110\overline{011001102}\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	IIIe
$=$	$10000001\mathbf{2}00111100\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$10000001\mathbf{2}00111100\mathbf{2}00011000\overline{000110002001111002}\mathbf{2}01111110\mathbf{2}$	
	$00011000\mathbf{2}10100101\mathbf{2}$	IIa
$=$	$10000001\mathbf{2}01111110\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$10000001\mathbf{2}01111110\overline{011111102}\mathbf{2}10000001\mathbf{2}01000010\mathbf{2}00011000\mathbf{2}$	
	$10100101\mathbf{2}$	IIIa
$=$	$10000001\mathbf{2}10000001\mathbf{2}01000010\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$10000001\mathbf{2}10000001\mathbf{2}01000010\overline{010000102}\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}$	
	$00011000\mathbf{2}10100101\mathbf{2}$	IIIb
$=$	$10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}00011000\overline{000110002101001012}\mathbf{2}$	
	$00100100\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	IIb
$=$	$10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}00100100\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}00100100\mathbf{2}10100101\overline{101001012001001002}\mathbf{2}$	
	$00011000\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	IIc
$=$	$10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}00011000\overline{000110002100110012}\mathbf{2}10011001\mathbf{2}$	
	$10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	IId
$=$	$10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$10000001\overline{10000001}x10000001\mathbf{2}10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	IVa
$=$	$x10000001\mathbf{2}10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$x10000001\overline{10000001}x10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	IVa
$=$	$xx10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$xx10011001\overline{10011001}y10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	IVb
$=$	$xxy10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$xxy10011001\overline{10011001}y10100101\mathbf{2}10100101\mathbf{2}$	IVb
$=$	$xxyy10100101\mathbf{2}10100101\mathbf{2}$	redukce
$\Rightarrow$	$xxyy10100101\overline{10100101}z10100101\mathbf{2}$	IVc
$=$	$xxyyz10100101\mathbf{2}$	redukce
$\Rightarrow$	$xxyyz10100101\overline{10100101}z$	IVc
$=$	$xxyyzz$	redukce

Výše uvedená derivace je na první pohled mnohem složitější. Přehlednost ubírá zejména kódování nonterminálních symbolů binárními řetězci. Vzhledem k tomu, že každý nonterminální symbol z původní gramatiky G je v Γ reprezentován binárním řetězcem o délce osm symbolů, zvyšuje se výrazným způsobem i prostorová složitost. Nesporné však je, že derivace byla provedena pouze s využitím bezkontextových pravidel. Důležitou roli v jejich aplikaci hraje volbá správného symbolu **2***, který má být přepsán. Jedině tak je zajištěna korektní redukce vzniklých inverzních podřetězců. Pokud by byl pro přepis zvolen nesprávný symbol, vzniklý inverzní podřetězec by nemohl být redukcí zcela odstraněn a z takto získané větné formy by již nebylo možné vygenerovat platnou větu jazyka.*

Kapitola 5

Možnosti praktického uplatnění studovaných modelů

Přestože patří studovaná témata spíše do teoretické oblasti, zkusme se v této kapitole zamyslet nad případnými možnostmi praktického uplatnění představených formálních modelů. Každému takovému uplatnění však musí nutně předcházet výzkum týkající se nalezení vhodné oblasti pro uplatnění a použitého přístupu pro praktické nasazení. Berme tedy tuto kapitolu pouze jako náčrt možných aplikací a inspiraci pro další výzkum.

Studované formální modely můžeme podle zjištěných vyjadřovacích schopností rozdělit do dvou skupin. První skupinu budou tvořit oboustranné zásobníkové automaty a bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálů, které generují třídu rekurzívně vyčíslitelných jazyků. Ve druhé skupině budou vertikálně omezené gramatiky, jejichž generativní síla pokrývá pouze třídu regulárních jazyků.

Oblast teoretické informatiky a formálních jazyků poskytuje silný matematický základ pro obor překladačů. Jako první se tedy nabízí uplatnění právě v překladačích. Bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálů mohou být použity jako prostředek pro syntaktickou analýzu shora dolů, kdy postupujeme od startovacího nonterminálu a snažíme se najít způsob (pořadí a místa aplikací jednotlivých přepisovacích pravidel) vedoucí k vygenerování požadované věty jazyka. Velkou výhodou těchto gramatik je skutečnost, že obsahují pouze bezkontextová přepisovací pravidla. Binární zakódování jednotlivých nonterminálních symbolů navíc poskytuje možnost využít nejnižší jednotky pro zobrazení v počítačích—bity—a tím redukovat paměťovou náročnost pro uchování jednotlivých větných forem. Nevýhodou je zde nedeterministický přístup. Konstrukce syntaktického analyzátoru by mohla být proto nejprve zaměřena na takové jazyky, které je možné generovat deterministicky. S tím však souvisí nalezení algoritmu, který by určoval aplikovatelné přepisovací pravidlo a současně i pozici nonterminálního symbolu v aktuální větné formě, který by byl tímto pravidlem přepsán a jehož přepisem by nedošlo k zablokování derivace z důvodu nemožnosti redukce vzniklých inverzních podřetězců. Takovýto algoritmus by musel být založen na průchodu celou větnou formou, neboť každé binární kódové slovo jednoho nonterminálního symbolu v původní gramatice je od ostatních odděleno symbolem 2, který jako jediný může být ve větné formě přepsán. Počet kódových slov ve větné formě je tedy shodný s počtem těchto dvojek, které mohou být přepsány a tudíž v každém derivačním kroku je minimálně právě tolik možností aplikace libovolného přepisovacího pravidla. Většina aplikací však nemá smysl právě z důvodu nemožnosti následné redukce vzniklých inverzních podřetězců. Otázkou však zůstává, zda tento algoritmus svoji časovou složitostí nezatíží celý syntaktický analyzátor nad přijatelnou mez.

Uplatnění těchto gramatik však nemusí být pouze v teorii formálních jazyků, nebo v oblasti překladačů. Cílem nemusí být ani generování konkrétního jazyka. V některých případech může být žádoucí provádět prostou simulaci vývoje nebo růstu organismů, krystalů, a podobně. V chemii by mohlo být přínosné použít tyto gramatiky pro simulaci různých chemických reakcí na úrovni molekul případně sloučenin vzájemně se ovlivňujících. Inverzní vlastnost volné grupy by zde zajišťovala např. neutralizační reakci dvou látek, k jejichž spojení by v dané struktuře došlo. Vzhledem k zákonu zachování energie však není možné, aby dva prvky zcela vymizely jako je tomu v našich gramatikách. Můžeme však uvažovat, že se např. kapalné skupenství určitého prvku mění na plynné, případně vzniká jiná forma energie, jako je např. teplo. Změna tohoto skupenství by byla simulována právě redukcí na úrovni volné grupy tak, že nové skupenství nebo vzniklé teplo by v další části simulace již nebylo uvažováno. Konkrétní reprezentace takovýchto experimentů by však již záležela na chemicích, kteří by mohli tyto gramatiky přizpůsobit svým požadavkům.

Další formální model—oboustranné zásobníkové automaty a jejich varianta s redukovaným počtem zásobníkových symbolů—by mohly nalézt uplatnění pro syntaktický analyzátor zdola nahoru. Podobně jako v předchozím případě by i zde mohlo být využití binárních kódů jednotlivých symbolů na zásobníku směřováno na co největší úsporu paměti využitím bitové reprezentace obsahu zásobníku. V tomto případě by bylo jednodušší omezit se pouze na deterministické varianty automatů. Otázkou zůstává, jak by byla tímto omezením změněna vyjadřovací síla. Je však naděje, že uplatnění oboustranných zásobníkových automatů pro syntaktickou analýzu může zvýšit schopnosti analyzátoru postaveného na běžném zásobníkovém automatu. Pro zjištění konkrétních vlastností však bude nutný další výzkum, který již přesahuje možnosti této práce.

Nyní se zkusme zamyslet nad možnostmi praktického uplatnění posledního formálního modelu (vertikálně omezené gramatiky v Kurodově normální formě), který má však nejnižší vyjadřovací shopnosti ze všech modelů studovaných v této práci. Jak bylo dokázáno, síla gramatik omezených popsaným vertikálním způsobem byla radikálně snížena až na úroveň regulárních jazyků. Princip činnosti těchto gramatik je však ve srovnání s ostatními modely se stejnými vyjadřovacími schopnostmi (regulární gramatiky, konečné automaty) poměrně složitý a pro praktické uplatnění zdánlivě nepoužitelný. Vzhledem k tomu, že jsme přesně určili vyjadřovací schopnosti gramatik omezených tímto způsobem, můžeme je použít pro dokazování vlastností jiných jazyků. Pokud bychom pro daný jazyk zkonstruovali gramatiku v Kurodově normální formě s těmito vlastnostmi, zároveň tím dokážeme, že je uvedený jazyk regulární. To může být výhodné zejména tam, kde je daný jazyk natolik složitý, že by přímá konstrukce konečného automatu případně regulární gramatiky byla neúměrně náročná. Sestrojením vertikálně omezené gramatiky v Kurodově normální formě dokážeme, že je daný jazyk skutečně regulární a podle uvedené konstrukce můžeme zkonstruovat i konečný automat, který jej přijímá.

Poznamenejme, že výše uvedené skutečnosti představují velmi hrubé náčrty možných praktických aplikací. Nesmíme však zapomenout, že k jejich realizaci může vést další náročný výzkum. Uvedené nápady však mohou posloužit jako vhodná inspirace pro další bádání.

Kapitola 6

Závěr

V současné teorii formálních jazyků existuje nesčetné množství různých formálních modelů pro popis jazyků. Některé jsou staré několik desítek let a staly se již jakýmisi základními modely pro popis zejména programovacích jazyků. Z těchto modelů však stále ještě vznikají další různé varianty, které zavádějí určité další prvky do těchto modelů, případně je různými způsoby modifikují. Většina těchto modifikací je motivována zvýšením vyjadřovací síly původního modelu.

My na tento trend částečně navazujeme a představujeme další nové modely, které jsou modifikací některých modelů stávajících. Shrňme proto na závěr hlavní výsledky této práce, pokusme se nastínit některé další směry výzkumu v této oblasti a rovněž i možnosti praktického uplatnění.

6.1 Oboustranné zásobníkové automaty

Běžný zásobníkový automat jsme rozšířili o možnost odebírat a vkládat symboly z obou jeho stran. Zásobník byl tímto změněn na oboustrannou frontu, ke které však přistupujeme specifickým způsobem v tom smyslu, že v každém kroku pracujeme současně s oběma jejími konci. Tím jsme do jisté míry rozšířili paměťové schopnosti běžného zásobníkového automatu. Jak jsme se již zmínili dříve, má tento přístup několik výhod. Zejména je to celistvost takto definovaného zásobníku. To nám otevírá možnost definovat zásobník např. nad volnou grupou a tím využít zejména vlastnosti inverzních prvků. Další výhodou je to, že můžeme takovýto zásobník jednoduše transformovat na (jednostrannou) frontu (pouhým omezením tvarů přechodových pravidel). Vhodným kódováním zásobníkových symbolů je pak možné redukovat zásobníkovou abecedu na pouhé čtyři symboly a tím částečně snížit i počet přechodových pravidel. Důkazy týkající se této problematiky potvrzují, že automaty tohoto typu mají stejné vyjadřovací schopnosti jako má Turingův stroj. Jinými slovy, jsou schopny definovat celou třídu rekurzívně vyčíslitelných jazyků. Kromě toho přijímají každou větu jazyka s jedinou obrátkou zásobníku.

6.2 Vertikální kontext v obecných gramatikách

V Chomského gramatikách je každý derivační krok prováděn pouze s ohledem na kontext v aktuální větné formě. My zavádíme zcela nový pohled na derivační proces jako celek, kdy sledujeme kromě běžného kontextu v jediné větné formě (nazvěme jej horizontální) i kontext mezi dříve vygenerovanými větnými formami—jakýsi vertikální kontext. V tomto smyslu zavádíme určitá omezení na běžné gramatiky v Kurodově normální formě a studujeme vliv použitých omezení na generativní schopnosti těchto gramatik. S ohledem na omezení maximální šířky segmentu a maximálního počtu kontextových derivací na hranici jsme vytvořili nový formální model založený na obecných gramatikách v Kurodově normální formě.

Dokázali jsme, že takováto omezení degradují gramatiky v Kurodově normální formě až na úroveň regulárních jazyků, tedy na samé dno Chomského hierarchie.

Zkusme nyní naznačit některé další možné směry výzkumu vertikálního kontextu v obecných gramatikách. Součástí studia vertikálně omezených gramatik bylo i studium dalšího typu vertikálního omezení, které však, jak se na samém konci ukázalo, nevedlo k požadovanému cíli. Jako výchozí byly opět použity gramatiky v Kurodově normální formě. Maximální šířka sloupce byla taktéž omezena stejným způsobem jak bylo popsáno v kapitole 5. Žádným způsobem však nebyl omezen počet aplikací jednotlivých kontextových přepisovacích pravidel na hranicích sloupců. Místo toho bylo omezeno jejich pořadí tak, že všechny aplikace kontextových přepisovacích pravidel na hranici i musely být provedeny dříve než kontextové aplikace na hranici $i-1$ pro $i = 2, 3, \ldots$. Původní domněnka byla ta, že takto omezené gramatiky generují nekonečnou hierarchii jazyků, jejíž jednotlivé části jsou dány právě počtem sloupců, který je obsažen v každé derivaci. Jako prostředek pro dokázání této skutečnosti byly zvoleny tzv. zásobníkové automaty s možností reverzace zásobníku popsané v [6] a [13]. Přestože byl vytvořen algoritmus konstrukce zásobníkového automatu tohoto typu pro danou gramatiku, další důkaz již nevedl k potvrzení domněnky. Jako otevřený problém zde proto formulujme myšlenku, že gramatiky v Kurodově normální formě omezené tímto nebo podobným vertikálním způsobem pravděpodobně mohou generovat nekonečnou hierarchii jazyků shodnou s hierarchií jazyků generovanou zásobníkovými automaty s možností reverzace zásobníku.

6.3 Bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálů

Hlavní modifikace v tomto případě spočívala v zavedení relace přímé derivace nad volnými grupami místo nad volnými monoidy, jak bývá obvyklé. Klíčovou úlohu v celém derivačním procesu hrají inverzní symboly volné grupy, které umožňují implicitní vymazávání z větných forem bez použití kontextových přepisovacích pravidel. Pomocí vhodné konstrukce gramatiky tím odpadá nutnost použití vymazávacích kontextových přepisovacích pravidel tvaru $AB \rightarrow \varepsilon$, které jsou jinak nezbytné. Podobným námětem se zabývají i autoři [9] nebo [5]. V jejich publikacích však pro zachování síly na úrovni rekurzívně vyčíslitelných jazyků musí využít i tato kontextová přepisovací pravidla. Jejich hlavní cíl je spíše orientován na redukci nonterminálních symbolů případně složitějších kontextových pravidel, která se snaží nahradit omezeným počtem těch nejjednodušších—např. již zmíněnými vymazávacími.

My jsme však v našich gramatikách dosáhli obojího. Nejen že jsme vhodným kódováním zredukovali potřebný počet nonterminálních symbolů, ale dosáhli jsme i úplného odstranění kontextových přepisovacích pravidel. Každý rekurzívně vyčíslitelný jazyk tedy může být generován bezkontextovou gramatikou, jejíž derivace jsou definovány nad volnou grupou generovanou její úplnou abecedou, která obsahuje právě osm nonterminálů.

Literatura

[1] M. Berka. Zásobníkový automat nad volnou grupou, 2006. [diplomová práce].

[2] R. Bidlo. Context-free grammars over free groups. In *Proceedings of 8th International Conference ISIM'05 Information System Implementation and Modeling*, pages 95–100, 2005.

[3] R. Bidlo, P. Blatný, and A. Meduna. Context-free and e0l derivations over free groups. *Schedae Informaticae*, 2007. in press.

[4] P. Blatný. *Formální modely nad volnými grupami*. PhD thesis, Falulty of Information Technology BUT, 2007.

[5] V. Geffert. How to generate languages using only two pairs of parentheses. *J. Inform. Process. Cybernet*, 1991.

[6] M. Holzer and M. Kutrib. Flip-pushdown automata: k+1 pushdown reversals are better than k. In *J.C.M. Baeten et al. (Eds.): ICALP 2003, LNCS 2719*, pages 490–501. Springer-Verlag, Berlin, 2003.

[7] H. C. M. Kleijn and G. Rozenberg. On the generative power of regular pattern grammars. *Acta Informatica*, 20:391–411, 1983.

[8] D. Kolář and A. Meduna. Regulated pushdown automata. *Acta Cybernetica*, 2000(4):653–664, 2000.

[9] D. Kolář and A. Meduna. Homogenous grammars with a reduced number of non-context-free productions. *Information Processing Letters*, 2002(81):253–257, 2002.

[10] A. Meduna. Simultaneously one-turn two-pushdown automata. *International Journal of Computer Mathematics*, 2003(82):1–9, 2003.

[11] A. Meduna. *Automata and Languages: Theory and Applications [Springer, 2000]*. Springer Verlag, 2005.

[12] G. Rozenberg and A. Salomaa, editors. *Handbook of formal languages, vols. 1-3*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[13] P. Sarkar. Pushdown automaton with the ability to flip its stack. *Electronic Colloquium on Computational Complexity (ECCC)*, 1(081), 2001.

Životopis autora

Jméno:	Radek Bidlo
Narozen:	2.11.1980
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
	cvičení v předmětu Základy programování
	vedení bakalářských prací
	recenze bakalářských a diplomových prací
	cvičení v předmětu Úvod do softwarového inženýrství
Publikace:	Spoluautor tří publikací v mezinárodních časopisech
	Autor/spoluautor příspěvků na konferencích
	Student EEICT 2003, 2004, 2005, 2006
	Autor příspěvku na mezinárodní konferenci ISIM 2005
	Spoluautor dvou příspěvků na konferenci MEMICS 2005
	Spoluautor příspěvku na konferenci MEMICS 2006
	Spoluautor příspěvku na konferenci WFM 2007