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

DIZERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE Ing. RADEK BIDLO
AUTHOR

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

O SÍLE NĚKTERÝCH MODIFIKOVANÝCH FORMÁLNÍCH MODELŮ

ON THE POWER OF SOME MODIFIED FORMAL MODELS

DIZERTAČNÍ PRÁCE
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.

Citace

Radek Bidlo: O síle některých modifikovaných formálních modelů, dizertační práce, Brno, FIT VUT v Brně, 2007

Prohlášení

Prohlašuji, že jsem tuto dizertační práci vypracoval samostatně pod vedením školitele, prof. RNDr. Alexandera Meduny, CSc. Některé výsledky byly dosaženy společně s kolegou Ing. Petrem Blatným 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.

. . . .
Radek Bidlo
19. května 2007

Poděkování

Na tomto místě bych rád poděkoval svému školiteli, prof. RNDr. Alexanderu Medunovi, CSc., za jeho inspiraci a nápady v průběhu celého výzkumu. Dále bych chtěl poděkovat kolegovi, Ing. Petru Blatnému, za spolupráci při výzkumu, psaní článků, konferenčních příspěvků a prezentací.

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. Uveďme si nyní některé možné klasifikace formálních jazyků a modelů pro jejich popis.

1.1 Klasifikace jazyků a formálních modelů

V roce 1956 rozdělil americký jazykovědec Avram Noam Chomsky jazyky do hierarchie podle tvarů přepisovacích pravidel gramatik, kterými mohou být generovány. Tato hierarchie byla jedním z nejvýznamnějších objevů dvacátého století v oblasti teorie formálních jazyků a dosud nese jeho jméno. Přestože se postupem času objevily další formální modely, které svými vyjadřovacími schopnostmi zasahují přes několik tříd jazyků Chomského hierarchie (jmenujme například některé typy L-systémů, které jsou schopné popsat z každé skupiny pouze některé jazyky), jedná se stále o jedno ze základních členění jazyků a gramatik.

Uveďme si nyní přehledně jednotlivé třídy Chomského hierarchie jazyků včetně základních formálních modelů, kterými je možné tyto jazyky popsat.

- Jazyky typu 0 (rekurzívně vyčíslitelné jazyky) — zahrnují všechny jazyky s gramatickým základem. Základní formální modely, které jsou schopny popsat tyto jazyky, jsou obecné (neomezené) gramatiky a Turingovy stroje.

- Jazyky typu 1 (kontextové jazyky) — základní formální modely, kterými můžeme tyto jazyky definovat, jsou kontextové gramatiky a lineárně ohraničené automaty.

- Jazyky typu 2 (bezkontextové jazyky) — tyto jazyky představují určitý teoretický základ syntaxe mnoha programovacích jazyků. Základní formální modely pro tento typ jazyků jsou zejména bezkontextové gramatiky a zásobníkové automaty.

- Jazyky typu 3 (regulární jazyky) — nejjednodušší jazyky Chomského hierarchie. Zahrnují rovněž i všechny konečné jazyky a jejich základní formální modely jsou regulární gramatiky a konečné automaty.

Poznamenejme, že v dalším výkladu budeme pro označetní těchto tříd jazyků používat pojmy rekurzívně vyčíslitelné jazyky, kontextové, bezkontextové a regulární jazyky. Tyto třídy jazyků jsou generovány postupně obecnými, kontextovými, bezkontextovými a regulárními gramatikami. V některé literatuře se však můžeme setkat též s označením pomocí čísla typu. Výše uvedená klasifikace podle Chomského rozděluje jazyky od nejjednodušších (tj. třída regulárních jazyků včetně všech konečných jazyků) až po nejsložitější, které ještě lze algoritmicky popsat (třída rekurzívně vyčíslitelných jazyků). Přitom platí, že každý jazyk typu i, kde $i <= 3$ je zároveň jazykem typu k, kde $0 \leq k < i$. Chomského klasifikace je v teorii formálních jazyků široce využívána při dokazování vyjadřovacích schopností dalších formálních modelů, kdy je vždy snaha jednoznačně specifikovat třídu jazyků, kterou je konkrétní model schopen popsat.

V současné době existuje velké množství dalších formálních modelů, které se od sebe liší jak způsobem, kterým jazyky definují, tak i vyjadřovacími schopnostmi. Formální modely, které jsme uvedli výše u každé třídy jazyků Chomského hierarchie, nebyly zvoleny zcela náhodně. Vždy jsme uvedli jeden model z oblasti gramatik a jeden z oblasti automatů. Kromě klasifikace modelů podle jejich vyjadřovacích schopností (zpravidla vztažená k Chomského hierarchii jazyků) můžeme tedy tyto modely rozčlenit i podle dalších hledisek— jedním z nich je právě členění na gramatiky a automaty.

- Generativní modely (gramatiky) — typickými představiteli formálních modelů této kategorie jsou gramatiky. Jak již vyplývá z přívlastku *generativní*, tyto modely jednotlivé věty jazyka generují. Jinými slovy, jazyk je popsán množinou všech vět, které může daný model (gramatika) generovat z nějakého výchozího axiomu. K tomuto účelu obsahují gramatiky prostředky, kterými tento axiom dále rozvíjí až k výsledné větě jazyka.

- Akceptační modely (automaty) — automaty tvoří další významnou třídu formálních modelů pro popis jazyků. Na rozdíl od gramatik však definují jazyky množinou všech řetězců, které jsou schopny přijmout ze své vstupní pásky, a zároveň splnit podmínky pro přijetí daného řetězce. Zjednodušeně můžeme říci, že akceptační modely postupují opačným způsobem než modely generativní.

- Ostatní modely — nepatří jednoznačně do žádné z dvou výše uvedených tříd a pro popis jazyků využívají jiné algebraické nebo matematické prostředky.

Pokud při členění formálních modelů pro popis jazyků vezmeme částečně v úvahu i 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 [27], bezkontextové gramatiky nad volnými grupami [2] (všechny tyto modely jsou modifikací bezkontextové gramatiky), vícezásobníkové automaty [23], zásobníkové automaty nad volnými grupami [1], řízené zásobníkové automaty [18] (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 [17], složitější typy L-systémů [24], gramatické systémy [26], [27], atd.

Jak je zřejmé z uvedených členění, liší se tyto modely nejen svými vyjadřovacími schopnostmi, ale také způsoby, jakými formální jazyky definují. Některé formální modely byly vyvinuty na základě již známých modelů (např. základních) a hlavním cílem bylo vytvořit takovou modifikaci, která by zvýšila vyjadřovací schopnosti původního modelu. Jiné naopak vznikly zcela nezávisle na ostatních.

1.2 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 oblasti automatů budeme vycházet z běžného zásobníkového automatu. V oblasti gramatik se zaměříme na obecné gramatiky v Kurodově normální formě [20] a na bezkontextové gramatiky. Zejména nás bude zajímat vliv zavedených modifikací na vyjadřovací sílu konkrétního formálního modelu. Poznamenejme, že naším cílem v tomto případě nebude nutně zvýšení vyjadřovacích schopností daného modelu. To se týká zejména obecných gramatik.

Představme si na tomto místě stručně a neformálně modely, které zavedenými modifikacemi vytvoříme.

- Oboustranné zásobníkové automaty — běžný zásobníkový automat obsahuje zásobník jako vnější paměť teoreticky neomezené velikosti. V této paměti je v každém okamžiku přístupný pouze symbol, který je na samém vrcholu zásobníku. K ostatním symbolům nelze přistoupit dříve, dokud se neobjeví právě na vrcholu, tedy dokud nejsou všechny symboly, které leží nad ním, ze zásobníku odstraněny. Jak jsme si již uvedli dříve, zásobníkové automaty definují třídu bezkontextových jazyků.

  V našem výzkumu provedeme velmi jednoduchou modifikaci zásobníkového automatu. Ta bude spočívat v tom, že do zásobníku umožníme přístup z obou stran. Poznamenejme, že touto modifikací vytvoříme ze zásobníku datovou strukturu, která se běžně nazývá oboustranná fronta. My však pro naše účely budeme dále používat název oboustranný zásobník, aby bylo zřejmé, že jsme na počátku vycházeli ze zásobníkového automatu. V dalším textu předložíme rigorózní důkaz, který se týká vyjadřovacích schopností oboustranných zásobníkových automatů. 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. Ten podrobně popíšeme dále. Hlavním cílem bude opět studium jeho vyjadřovacích schopností.

- Bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálů — v gramatikách Chomského hierarchie je relace přímé derivace definována nad volnými monoidy generovanými úplnými abecedami těchto gramatik. Připomeňme, že úplnou abecedou je nazýváno sjednocení množiny terminálních a nonterminálních symbolů. Obě tyto množiny musí být konečné. Počet jejich symbolů však není nijak omezen.

  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. Tím vytvoříme nový druh bezkontextových gramatik a budeme zkoumat, jak zavedené modifikace ovlivní jejich generativní sílu.

1.3 Struktura textu dizertační práce

Celá práce je rozdělena do osmi kapitol. Po neformálním úvodu následuje druhá kapitola, která představuje úvod do teorie formálních jazyků a definuje nezbytné algebraické pojmy, jejichž znalost je vyžadována pro pochopení obsahu dalších částí.

Třetí kapitola shrnuje současný stav poznání v oblasti formálních modelů pro popis jazyků. Popisuje nejrozšířenější verze gramatik a automatů, představuje některé méně známé varianty využívané v dalších sekcích a neformálně se zmiňuje i o vybraných modifikacích nejznámějších modelů, které v průběhu uplynulých desetiletí vznikly.

V dalších třech kapitolách jsou již prezentovány nové výsledky, které tvoří hlavní jádro této dizertační práce.

Čtvrtá kapitola představuje oboustranné zásobníkové automaty včetně varianty s redukovaným počtem symbolů zásobníkové abecedy, jejich formální definici a princip činnosti. Dále je zde uveden rigorózní důkaz týkající se vyjadřovacích schopností těchto automatů.

Pátá kapitola zavádí pojem vertikálního kontextu v obecných gramatikách. Po definici nezbytných pojmů je formálně popsána věta týkající se vyjadřovacích schopností vertikálně omezených gramatik a představen rigorózní důkaz, který demonstruje její platnost.

V šesté kapitole jsou představeny bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálních symbolů. Je uvedena formální definice, princip činnosti a formální důkaz týkající se jejich vyjadřovacích schopností.

Všechny kapitoly o nově vzniklých formálních modelech jsou členěny pokud možno jednotným způsobem a jsou zejména zaměřeny na

- formální definici
- formální popis principu činnosti
- formulaci hypotézy týkající se vyjadřovacích schopností
- rigorózní důkaz potvrzující uvedenou hypotézu
- shrnutí dosažených výsledků

Sedmá kapitola nastiňuje možné praktické aplikace studovaných formálních modelů a v poslední—osmé—kapitole jsou shrnuty dosažené výsledky a diskutovány možné směry dalšího výzkumu v této oblasti.

Kapitola 2

Algebraické základy teorie formálních jazyků

Tato kapitola obsahuje definice elementárních algebraických pojmů, jejichž znalost je nezbytná pro pochopení dalších diskutovaných problémů. Zde uvedené informace lze též najít např. v [14], [16], [23] a [28], ale i v mnoha dalších publikacích, kterých je na toto téma k dispozici velmi mnoho.

2.1 Množiny

Jako první zopakujeme pojem množina a připomeneme základní operace, které budeme v dalším textu používat.

Definice 2.1 — množina, kardinalita množiny Pojem *množina* je chápán intuitivně jako soubor prvků (objektů) seskupených v jeden celek. O každém prvku přitom můžeme jednoznačně prohlásit, zda do množiny patří, či nikoliv. Význačné postavení mezi množinami má množina *prázdná*, která neobsahuje žádné prvky. Označuje se symbolem $\emptyset$. Je-li A množina a a prvek této množiny, potom můžeme příslušnost prvku a do množiny A vyjádřit symbolicky jako

$$a \in A.$$

Poznamenejme, že definice klasické množiny připouští, aby se v ní každý prvek vyskytoval nejvýše jednou.

Kardinalitou množiny A, $card(A)$, nazveme číslo udávající počet prvků v množině. Nechť

$$A = \{a_1, a_2, \ldots, a_n\}$$

je množina obsahující prvky $a_1, a_2, \ldots, a_n$. Potom:

1. $card(A) = 0$, pokud $n = 0$ (prázdná množina, $\emptyset$)
2. $card(A) = n$, pokud $n \geq 1$

Definice 2.2 — sjednocení množin Nechť A a B jsou množiny. *Sjednocením* těchto množin označíme množinu

$$A \cup B = \{a | a \in A \text{ nebo } a \in B\}$$

Definice 2.3 — průnik množin Nechť A a B jsou množiny. *Průnikem* těchto množin označíme množinu

$$A \cap B = \{a | a \in A \text{ a zároveň } a \in B\}$$

Definice 2.4 — rozdíl množin Nechť A a B jsou množiny. *Rozdílem* množiny A a množiny B rozumíme množinu

$$A - B = \{a | a \in A \text{ a zároveň } a \notin B\}$$

Definice 2.5 — rovnost množin Nechť A a B jsou množiny. Tyto množiny prohlásíme za *sobě rovné*, píšeme

$$A = B,$$

pokud pro každý prvek $a \in A$ platí $a \in B$ a zároveň pro každý prvek $b \in B$ platí $b \in A$.

Definice 2.6 — podmnožina množiny, potenční množina Nechť A a B jsou množiny. Řekneme, že množina A je *vlastní podmnožinou* množiny B, píšeme

$$A \subset B,$$

jestliže pro každý prvek $a \in A$ platí $a \in B$ a zároveň $A \neq B$. Dále řekneme, že A je *podmnožinou* množiny B, píšeme

$$A \subseteq B,$$

jestliže pro každý prvek $a \in A$ platí $a \in B$, přičemž může platit i $A = B$.

Potenční množina množiny A je množina všech podmnožin množiny A definována jako

$$2^A = \{X | X \subseteq A\}.$$

2.2 Relace a zobrazení

S množinami úzce souvisí i následující pojmy, které tvoří algebraický základ pro většinu gramatik a automatů.

Definice 2.7 — kartézský součin Nechť A a B jsou množiny. Binárním *kartézským součinem* těchto množin (v uvedeném pořadí) rozumíme množinu všech uspořádaných dvojic definovanou jako

$$A \times B = \{(a, b) | a \in A \text{ a zároveň } b \in B\}.$$

Kartézský součin lze zobecnit pro libovolný počet množin. Nechť tedy $A_1, A_2, \ldots, A_n$ jsou množiny, $n > 1$. $n-$árním kartézským součinem rozumíme množinu všech uspořádaných $n-$tic definovanou jako

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n) | a_i \in A_i \text{ pro } i = 1, 2, \ldots, n\}.$$

Definice 2.8 — relace *Binární relací* $R \subseteq A \times B$ rozumíme libovolnou podmnožinu kartézského součinu dvou množin A a B. Binární relace je nejčastěji se vyskytující případ relace. V obecném případě definujeme $n-$ární relaci R_n pro libovolné $n > 1$ jako

$$R_n \subseteq A_1 \times A_2 \times \cdots \times A_n,$$

kde A_i pro $i = 1, 2, \ldots, n$ jsou množiny.

Definice 2.9 — zobrazení *Zobrazení*

$$f : A \to B$$

je předpis, který každému prvku z množiny A jednoznačně přiřazuje právě jeden prvek z množiny B. Množina A se pak nazývá *definiční obor* (doména) a množina B *obor hodnot* (kodoména). Skutečnost, kdy je prvek $a \in A$ zobrazen na nějaký prvek $b \in B$ zapisujeme formálně jako

$$b = f(a).$$

Zobrazení se nazývá *injektivní*, pokud pro libovolné $a_1, a_2 \in A$, kde $a_1 \neq a_2$, platí

$$f(a_1) \neq f(a_2).$$

Jinými slovy, libovolné dva různé vzory z A mají v B různé obrazy. Není tedy možné, že by dva různé vzory z A měly stejný obraz v B, nebo jeden prvek v B měl dva nebo více vzorů v A.

Zobrazení nazveme *surjektivní*, pokud pro každé $b \in B$ existuje nějaké $a \in A$ takové, že

$$b = f(a).$$

Jinými slovy, každý prvek v B má nějaký vzor v A.

Zobrazení se nazývá *bijektivní*, pokud je injektivní i surjektivní.

2.3 Abecedy, řetězce a formální jazyky

Další definice vychází z některých dříve zavedených pojmů, jsou však již více orientovány do oblasti teorie formálních jazyků.

S většinou pojmů, které budeme definovat v této kapitole, jsme se již setkali v předchozích podkapitolách, kde jsme je však chápali intuitivně. Uveďme si tedy konečně jejich formální definici.

Definice 2.10 — abeceda *Abeceda* je konečná neprázdná množina prvků, které nazýváme *symboly*.

Definice 2.11 — řetězec Nechť Σ je abeceda.

1. ε je *řetězec* nad abecedou Σ (tzv. *prázdný řetězec*)

2. jestliže x je *řetězec* nad abecedou Σ a $a \in \Sigma$ je libovolný symbol této abecedy, pak také xa je *řetězec* nad abecedou Σ

Definice 2.12 — délka řetězce Nechť Σ je abeceda a nechť x je řetězec nad touto abecedou. *Délka řetězce* x je označována jako $|x|$ a je definována následujícím způsobem:

1. jestliže $x = \varepsilon$, pak $|x| = 0$

2. jestliže $x = a_1 a_2 \ldots a_n$ pro nějaké $n \geq 1$, kde $a_i \in \Sigma$ pro všechna $i = 1, 2, \ldots, n$, potom $|x| = n$

Definice 2.13 — reverzace řetězce Nechť Σ je abeceda a nechť $w = a_1a_2 \dots a_n$, kde $a_i \in \Sigma$ pro $i = 1, 2, \dots, n$, je libovolný řetězec nad touto abecedou. *Reverzací* (otočením) řetězce w rozumíme řetězec

$$w^R = a_n \dots a_2a_1.$$

Poznamenejme, že reverzace prázdného řetězce je opět prázdný řetězec.

Definice 2.14 — konkatenace řetězců Nechť Σ je abeceda a nechť x a y jsou řetězce nad touto abecedou. *Konkatenací* (spojením) řetězce x s řetězcem y rozumíme řetězec

$$x \cdot y = xy.$$

Poznamenejme, že podle zavedených zvyklostí se operátor konkatenace $\cdot$ většinou vynechává a formálně tedy zapíšeme spojení řetězce x s řetězcem y jednoduše jako xy.

Definice 2.15 — iterace množiny Uvažujme libovolnou abecedu Σ. Symbolem Σ^* označíme množinu všech řetězců nad abecedou Σ. Dále definujme $\Sigma^+ = \Sigma^* - \{\varepsilon\}$ množinu všech řetězců nad abecedou Σ vyjma řetězce prázdného. Množina Σ^*, resp. Σ^+ se nazývá *iterace*, resp. *pozitivní iterace* množiny Σ.

Poznamenejme, že Σ^* je volný monoid generovaný abecedou Σ a operací konkatenace. Jeho jednotkovým prvkem je prázdný řetězec ε. Formální definici volného monoidu uvedeme později.

Definice 2.16 — formální jazyk Nechť Σ je abeceda a nechť

$$L \subseteq \Sigma^*.$$

Potom se L nazývá *formální jazyk* nad abecedou Σ. Pokud bude z kontextu zřejmé, že se jedná o formální jazyk, budeme někdy L nazývat jednoduše *jazykem* nad abecedou Σ.

Z definice formálního jazyka je zřejmé, že se jedná o běžnou množinu a tudíž je možné provádět s jazyky všechny dříve popsané množinové operace. Kromě toho jsou však pro oblast teorie formálních jazyků definovány operace konkatenace, doplněk, mocnina, iterace a reverzace. Formálně jsou tyto operace popsány následujícími definicemi. Uvažujme pro ně libovolnou abecedu Σ a nějaké jazyky L_1 a L_2 nad touto abecedou.

Definice 2.17 — konkatenace jazyků *Konkatenací* jazyků L_1 a L_2 rozumíme jazyk

$$L_1 \cdot L_2 = \{xy | x \in L_1 \text{ a } y \in L_2\}.$$

Definice 2.18 — doplněk jazyka *Doplňkem* jazyka L_1 rozumíme jazyk

$$dop(L_1) = \{x | x \in \Sigma^* - L_1\}.$$

Definice 2.19 — mocnina jazyka *$n-$tou mocninou* jazyka L_1 rozumíme jazyk L^n, který je definovaný následujícím způsobem:

$$L_1^0 = \{\varepsilon\}$$
$$L_1^n = L_1 \cdot L_1^{n-1}$$

Definice 2.20 — iterace jazyka *Iterací* jazyka L_1 rozumíme jazyk

$$L_1^* = \bigcup_{i \geq 0} L_1^i$$

Pozitivní iterací jazyka L_1 rozumíme jazyk

$$L_1^+ = \bigcup_{i \geq 1} L_1^i$$

Definice 2.21 — reverzace jazyka *Reverzací* jazyka L_1 rozumíme jazyk

$$L_1^R = \{w^R | w \in L_1\}.$$

2.4 Základní algebraické struktury

Základní algebraické struktury, které nás budou pro další výklad nejvíce zajímat, jsou *grupoidy*, *pologrupy*, *monoidy* a *grupy*. Monoidy a grupy dále zobecníme na *volné monoidy* a *volné grupy*, které bodou hrát klíčovou roli v části našeho dalšího výzkumu. Nejprve si však musíme definovat pojem *algebraické operace* a základní *axiomy algebraických operací*, které jsou pro pochopení dalších algebraických struktur nejdůležitější.

Poznamenejme, že náplň této kapitoly byla čerpána z [11], [21] a [25].

2.4.1 Finitární algebraické operace

Definice 2.22 — algebraická operace Nechť n je celé nezáporné číslo a M je množina. Jestliže je každé uspořádané $n-$tici prvků $a_1, a_2, \ldots, a_n \in M$ přiřazen jednoznačně určený prvek $a \in M$, řekneme, že na množině M je definována *$n-$ární algebraická operace* (označme ji $\circ$). Prvek a nazveme výsledkem operace $\circ$ na prvky $a_1, a_2, \ldots, a_n$ a budeme jej označovat jako $a = \circ(a_1 a_2 \ldots a_n)$. Číslo n se potom nazývá *aritou* operace $\circ$.

Podle arity 0, 1 a 2 definujeme tři nejvýznamnější algebraické operace, které pořadě nazveme *nulární*, *unární* a *binární*.

Definice 2.23 — nulární operace *Nulární operací* na množině M nazýváme zobrazení $\circ : M^0 \to M$, jehož výsledkem je právě jeden z prvků množiny M. Tento prvek nezávisí na volbě prvků z M a můžeme jej chápat jako vyčlenění některého význačného prvku z množiny M.

Definice 2.24 — unární operace *Unární operací* na množině M nazýváme zobrazení $\circ : M \to M$, které každému prvku $a \in M$ přiřazuje právě jeden prvek $\circ(a) \in M$.

Definice 2.25 — binární operace *Binární operací* na množině M nazýváme zobrazení $\circ : M^2 \to M$. Prvek $\circ(a, b)$, kde $a, b, \circ(a, b) \in M$ nazýváme kompozicí prvků a, b vzhledem k binární operaci na množině M.

Pro úplnost uveďme, že takto prezentovaný zápis je většinou využíván u arit větších než dva. V případě binární operace je nejčastěji používán tzv. *infixový* zápis, kdy je operátor vepsán mezi oba operandy. V tomto případě by měl tvar $a \circ b$.

2.4.2 Axiomy binárních operací

Na binární operace na dané množině klademe určité požadavky, podle kterých potom získáváme objekty s různými algebraickými vlastnostmi. Tyto objekty nazýváme *algebraickými strukturami.* Binární operace na určité množině splňující axiomy příslušné algebraické struktury nazýváme *operacemi* této struktury.

Nechť tedy $\circ$ je binární operace a necť M je množina. Základní axiomy, které nás budou v souvislosti s naší problematikou zajímat, jsou:

A_0 *Uzavřenost operace*: ke každé dvojici prvků $a, b \in M$ přiřazujeme prvek $c = a \circ b$. Platí $c \in M$.

A_1 *Asociativní zákon*: Operace $\circ$ splňuje asociativní zákon, pokud $(a \circ b) \circ c = a \circ (b \circ c)$, kde $a, b, c \in M$.

A_2 *Existence neutrálního prvku*: Existuje takový prvek $\varepsilon \in M$, pro který platí $\varepsilon \circ a = a \circ \varepsilon = a$ pro všechna $a \in M$.

A_3 *Existence inverzního prvku*: ke každému prvku $a \in M$ existuje tzv. inverzní prvek $\overline{a} \in M$ takový, že platí $a \circ \overline{a} = \overline{a} \circ a = \varepsilon$, kde $\varepsilon \in M$ je neutrální prvek operace $\circ$ na množině M podle axiomu A_2.

A_4 *Komutativní zákon*: Operace $\circ$ splňuje komutativní zákon, pokud $a \circ b = b \circ a$, kde $a, b \in M$.

2.4.3 Grupoidy, pologrupy, monoidy a grupy

Než si definujeme *volnou grupu*, která nás jakožto algebraická struktura bude v tomto výkladu zajímat nejvíce, musíme si definovat i struktury, ze kterých bude definice volné grupy vycházet.

Definice 2.26 — grupoid Algebraická struktura $(M, \circ)$ definována binární operací $\circ$ na množině M se nazývá *grupoid*, pokud operace $\circ$ splňuje axiom A_0 uzavřenosti operace.

Př.: Uvažujme množinu celých čísel $\mathcal{Z}$ a operaci odčítání nad touto množinou. Struktura $(\mathcal{Z}, -)$ je grupoid.

Definice 2.27 — pologrupa Algebraická struktura $(M, \circ)$ jejíž operace splňuje axiom A_0 uzavřenosti operace a asociativní zákon podle axiomu A_1 se nazývá *pologrupa*. Ekvivalentní název je též *asociativní grupoid.*

Př.: Uvažujme opět množinu celých čísel $\mathcal{Z}$ a operaci sčítání nad touto množinou. Struktura $(\mathcal{Z}, +)$ je pologrupa.

Definice 2.28 — monoid Algebraická struktura $(M, \circ, \varepsilon)$ jejíž operace splňuje axiomy A_0, A_1 a A_2 se nazývá *monoid* s neutrálním prvkem ε. Ekvivalentní název je též *pologrupa s neutrálním prvkem.*

Výběr neutrálního prvku můžeme chápat jako definici určité nulární operace na množině M, která dává jako svůj výsledek právě prvek $\varepsilon \in M$.

Př.: Struktura $(\mathcal{Z}, +)$ obsahující množinu všech celých čísel a operaci sčítání je zároveň monoidem. Jejím neutrálním prvkem je nula.

Definice 2.29 — grupa Algebraická struktura $(M, \circ, \varepsilon, ^{-1})$, která splňuje axiomy A_0, A_1, A_2 a A_3, se nazývá *grupa.* Pokud je splněn i axiom A_4, jedná se o *komutativní (abelovskou) grupu.*

Grupu můžeme chápat jako monoid obsahující ke každému prvku $a \in M$ prvek inverzní $\overline{a} \in M$. Inverzní prvky můžeme definovat pomocí unární operace $^{-1} : M \to M$.

Př.: Množina reálných čísel $\mathcal{R}$ s binární operací sčítání $+$ a unární operací záporné hodnoty $-$ je grupou. Neutrálním prvkem je nula. Inverzním prvkem ke každému $x \in \mathcal{R}$ je $-x$. Potom platí $x + (-x) = -x + x = 0$ a $x + 0 = 0 + x = x$.

Poznamenejme, že existují další algebraické struktury, které se liší počtem binárních operací, případně dalšími axiomy, které musí jejich operace splňovat. Operace mohou být v obecném případě $n-$ární a jejich arity mohou být navzájem různé. Tyto struktury však nevyužijeme a proto je ani nebudeme uvádět. Z výše uvedených struktur pro nás budou základem zejména monoidy a grupy, které pro účely našeho dalšího výzkumu zobecníme na volné monoidy a volné grupy.

V kapitole 2.2 jsme definovali pojem zobrazení mezi dvěma množinami. Obdobně lze definovat i zobrazení mezi dvěma algebraickými strukturami stejného typu. Takovéto zobrazení se pak nazývá *homomorfismus.* Uveďme si formální definici.

Definice 2.30 — homomorfismus Nechť A a B jsou dvě algebraické struktury stejného typu a nechť

$$\Phi : A \to B$$

je zobrazení mezi těmito strukturami. Toto zobrazení nazveme *homomorfismem*, jestliže pro každou definovanou operaci $\circ_A$ struktury A a $\circ_B$ struktury B platí pro každé $x_i \in A$

$$\Phi(\circ_A(x_1, x_2, \ldots, x_n)) = \circ_B(\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_n)),$$

kde $0 \leq i \leq n$.

Je-li zobrazení Φ injektivní, nazývá se rovněž *injektivní homomorfismus*, nebo krátce *monomorfismus.*

Je-li Φ surjektivní, nazývá se též *surjektivní homomorfismus*, nebo krátce *epimorfismus.*

Konečně je-li zobrazení Φ bijektivní (tj. injektivní a surjektivní), nazývá se *bijektivní homomorfismus*, nebo krátce *izomorfismus.*

Př.: Uvažujme množinu přirozených čísel $\mathcal{N}$ s operací sčítání a funkci

$$f(x) = 2x,$$

kde $x \in \mathcal{N}$. Funkce f je homomorfismem na množině přirozených čísel, neboť

$$f(a + b) = 2(a + b) = 2a + 2b = f(a) + f(b).$$

2.5 Volné monoidy a volné grupy

V této sekci si zavedeme pojem volné grupy, který nás bude zajímat nejvíce. S ohledem na předchozí definice monoidu a grupy vidíme, že grupa je struktura o něco složitější. Kromě axiomů pro monoid musí obsahovat ke každému svému prvku i prvek inverzní, což může činit potíže. Pojďme si tedy nejprve definovat strukturu jednodušší a sice volný monoid.

2.5.1 Volný monoid

Nechť je Σ abeceda. Uvažujme libovolné řetězce $w_1 = a_1a_2 \dots a_m$ a $w_2 = b_1b_2 \dots b_n$, kde $m \geq 0$ a $n \geq 0$. Pomocí asociativního operátoru konkatenace (spojení) $\cdot$ můžeme vytvořit řetězec $w = w_1 \cdot w_2 = a_1a_2 \dots a_m b_1 b_2 \dots b_n$ o délce $|w| = m + n$ symbolů.

Označme Σ^* množinu všech konečných řetězců nad abecedou Σ včetně prázdného řetězce ε společně s operátorem konkatenace $\cdot$. Potom Σ^* nazýváme *volným monoidem* nad abecedou Σ. Nazveme-li každý prvek $a \in \Sigma$ řetězcem o délce $|a| = 1$, stane se Σ podmnožinou Σ^*. Říkáme, že Σ *generuje* Σ^*.

Definice 2.31 — volný monoid Množinu Σ nazveme *volnou bází* monoidu Σ^*, jestliže Σ generuje Σ^* a každé zobrazení $p : \Sigma \to P$, kde P je monoid, je možné rozšířit na homomorfismus $q : \Sigma^* \to P$. Monoid Σ^* nazveme *volným*, pokud má alespoň jednu volnou bázi.

Př.: Uvažujme abecedu $A = \{a, b, c\}$ a operaci konkatenace $\cdot$. Volný monoid generovaný touto abecedou a operací konkatenace je množina řetězců tvaru

$$A^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, \dots\},$$

kde ε (prázdný řetězec) je neutrálním prvkem. Jsou-li u, v dva řetězce z A^*, pak rovněž i

$$u \cdot v \in A^*$$

a zároveň platí

$$u \cdot \varepsilon = \varepsilon \cdot u = u.$$

2.5.2 Volná grupa

Obdobně jako volný monoid budeme definovat i volnou grupu. Nechť Σ je abeceda. Definujme množinu Γ obsahující všechny symboly abecedy Σ a pro každé $a \in \Sigma$ bude obsahovat jeho inverzní protějšek $\overline{a}$, tedy

$$\Gamma = \{a, \overline{a} | a \in \Sigma\}.$$

Označme Γ° množinu všech konečných řetězců tvořených symboly z Γ. Tím jsme získali monoid s asociativním operátorem konkatenace. Zatím se však nejedná o grupu, neboť nemůžeme vyrušit případné výskyty dvojic symbolů $x\overline{x}$ a $\overline{x}x$. Definujme tedy, že pro každé $x, \overline{x} \in \Gamma$ platí $x\overline{x} = \overline{x}x = \varepsilon$, kde ε je prázdný řetězec a zároveň jednotkový prvek monoidu Γ°. Budeme-li každý prvek $a \in \Sigma$ chápat jako řetězec o délce $|a| = 1$, stane se Γ podmnožinou Γ°. Získáváme analogii s volným monoidem a vidíme, že Γ zjevně *generuje* Γ°.

Nyní zbývá vyřešit inverzní prvky. Zaveďme k tomu následující pojem. Libovolný řetězec z Γ° se nazývá *redukovaný*, pokud neobsahuje žádné výskyty tvaru $x\overline{x}$ a $\overline{x}x$. Vyjdeme-li z nějakého řetězce $w \in \Gamma^\circ$, můžeme provést konečný počet redukcí jednotlivých dvojic vzájemně inverzních symbolů x a $\overline{x}$. V okamžiku, kdy již nebude možné provést žádnou redukci, prohlásíme výsledný řetězec za *redukovaný tvar* řetězce w. Tento řetězec může být i prázdný, tedy ε. Přitom možností provedení jednotlivých redukcí může být velmi mnoho. Bez ohledu na pořadí provedení těchto redukcí však vždy dojdeme ke stejnému výsledku. To potvrzuje následující věta, jejíž důkaz lze najít v [25].

Věta 2.1 [25] Ke každému řetězci $w \in \Gamma^\circ$ existuje pouze jediný redukovaný řetězec.

Z dosud uvedených skutečností je zřejmé, že Γ° obsahuje různé řetězce, které mají společný redukovaný tvar. Definujme si proto ekvivalenci dvou řetězců.

Definice 2.32 — ekvivalentní řetězce Nechť $w_1, w_2 \in \Gamma^\circ$ jsou řetězce. Řekneme, že w_1 a w_2 jsou *ekvivalentní*, píšeme

$$w_1 \sim w_2,$$

pokud mají shodné redukované tvary. Toto je zjevně ekvivalenční relace.

Věta 2.2 [25] Konkatenací ekvivalentních řetězců získáme opět ekvivalentní řetězce, tedy

$$w \sim w', \quad v \sim v' \Rightarrow wv \sim w'v'$$

Všimněme si, že Γ° společně s operací konkatenace $\cdot$ a prázdným řetězcem ε je zjevně monoidem. Obsahuje však inverzní prvky, neboť pro libovolný řetězec $a_1a_2 \dots a_{n-1}a_n \in \Gamma^\circ$, kde $a_i \in \Gamma$ pro $i = 1, 2, \dots, n$ platí

$$a_1a_2 \dots a_{n-1}a_n \cdot \overline{a_n a_{n-1}} \dots \overline{a_2 a_1} = \varepsilon$$

Γ° nazveme tedy *volnou grupou* na Σ. Uveďme si pro úplnost i formální definici, která je analogická definici 2.31.

Definice 2.33 — volná grupa Množinu Γ nazveme volnou bází grupy Γ°, jestliže Γ generuje Γ° a každé zobrazení $p : \Gamma \to P$, kde P je grupa, je možné rozšířit na homomorfismus $q : \Gamma^\circ \to P$. Grupu Γ° nazveme *volnou*, pokud má alespoň jednu volnou bázi.

Př.: Příklad k volné grupě je pouze rozšířením příkladu volného monoidu o inverzní prvky, jejichž existence je vynucena definicí grupy. Uvažujme opět abecedu

$$A = \{a, b, c, \overline{a}, \overline{b}, \overline{c}\}$$

a operaci konkatenace $\cdot$. Volná grupa generovaná touto abecedou a operací konkatenace je rovněž množina řetězců tvaru

$$A^\circ = \{\varepsilon, a, b, c, \overline{a}, \overline{b}, \overline{c}, aa, ab, ac, a\overline{a}, a\overline{b}, a\overline{c}, \dots\},$$

kde ε (prázdný řetězec) je jejím neutrálním prvkem. Navíc je však symbol $\overline{a}$ inverzním symbolem k symbolu a, $\overline{b}$ je inverzním symbolem k symbolu b a $\overline{c}$ je inverzním symbolem k symbolu c. Pro zobecnění této skutečnosti na řetězce uvažujme řetězec u tvaru

$$u = x_1x_2 \dots x_n,$$

kde $x_i \in A$ *pro* $i = 1, 2, \ldots, n$. *Inverzním řetězcem k tomuto řetězci je pak řetězec*

$$\overline{u} = \overline{x_n} \ldots \overline{x_2 x_1}$$

a je zřejmé, že platí

$$u \cdot \overline{u} = \overline{u} \cdot u = \varepsilon.$$

Zároveň pro každý řetězec $u \in A^\circ$ *platí*

$$u \cdot \varepsilon = \varepsilon \cdot u = u.$$

Na závěr této kapitoly poznamenejme, že budeme volné grupy používat k modifikaci bezkontextových gramatik v kapitole 6 za účelem zvýšení jejich generativních schopností.

Kapitola 3

Gramatiky a automaty

Již v úvodu jsme neformálně představili Chomského hierarchii gramatik a jazyků, která rozděluje jazyky do několika tříd v závislosti na tvarech přepisovacích pravidel gramatik, kterými mohou být tyto jazyky generovány. Ke každé gramatice existuje zpravidla nějaký ekvivalentní (stejně silný) formální model v oblasti automatů. Pojďme si nejprve představit formální definici gramatiky a její varianty.

3.1 Gramatiky Chomského hierarchie

Jak jsme se již zmínili v úvodu, gramatiky můžeme též nazvat generátory jazyků, neboť jednotlivé věty každého jazyka jsou postupně generovány (vytvářeny) z nějakého startovacího symbolu (axiomu) systematickou aplikací tzv. přepisovacích pravidel.

3.1.1 Základní definice

Definice 3.1 — obecná gramatika *Obecná gramatika* (někdy je též označována jako neomezená gramatika, případně jenom gramatika) 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
- $N = V - T$ je abeceda nonterminálních symbolů
- P je konečná množina přepisovacích pravidel tvaru

  $$\alpha \to \beta,$$

  kde $\alpha \in V^*NV^*$ a $\beta \in V^*$
- $S \in N$ je startovací symbol

Poznamenejme, že čistě z algebraického hlediska je $P \subseteq V^*NV^* \times V^*$ konečná binární relace a kazdé přepisovací pravidlo $\alpha \to \beta$ je ve skutečnosti uspořádaná dvojice (α, β), kde $\alpha \in V^*NV^*$ a $\beta \in V^*$.

Definice 3.2 — kontextová gramatika Gramatika $G = (V, T, P, S)$ se nazývá *kontextová*, jestliže každé přepisovací pravidlo z P je tvaru

$$\alpha \to \beta,$$

kde $|\alpha| \leq |\beta|$, $\alpha \in V^*NV^*$ a $\beta \in V^*$.

Poznamenejme, že toto je jedna z možných definic. V některé literatuře je kontextová gramatika definována přepisovacími pravidly tvaru

$$\alpha A\beta \to \alpha\gamma\beta,$$

kde $\alpha, \beta \in V^*$, $A \in N$ a $\gamma \in V^+$.

V obou případech je možné přidat navíc následující podmínku. Jestliže $\varepsilon \in L(G)$ (viz Definice 3.8), pak je přípustné i pravidlo $S \to \varepsilon$ pod podmínkou, že se nonterminál S nevyskytne na pravé straně žádného přepisovacího pravidla.

Definice 3.3 — bezkontextová gramatika Gramatika $G = (V, T, P, S)$ se nazývá *bezkontextová*, jestliže každé přepisovací pravidlo z P je tvaru

$$A \to \alpha,$$

kde $A \in N$ a $\alpha \in V^*$.

Definice 3.4 — regulární gramatika Gramatika $G = (V, T, P, S)$ se nazývá *regulární*, jestliže každé přepisovací pravidlo z P je tvaru

$$A \to aB$$

nebo

$$A \to a,$$

kde $A, B \in N$ a $a \in T$. Jestliže $\varepsilon \in L(G)$ (viz Definice 3.8), pak je přípustné i pravidlo

$$S \to \varepsilon$$

pod podmínkou, že se nonterminál S nevyskytne na pravé straně žádného jiného přepisovacího pravidla.

3.1.2 Derivační proces

Následující definice jsou aplikovatelné na všechny výše uvedené gramatiky. Pojmem *gramatika* budeme mít tedy na mysli libovolnou gramatiku z výše uvedených (obecná, kontextová, bezkontextová, regulární). V této podkapitole si představíme princip, jakým každá gramatika generuje jednotlivé věty jazyka.

Definice 3.5 — přímá derivace Nechť $G = (V, T, P, S)$ je gramatika a nechť $\lambda, \mu \in V^*$ jsou řetězce takové, že

$$\lambda = \alpha\gamma\beta$$

a

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

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

$$\lambda \Rightarrow \mu[\gamma \to x]$$

nebo stručněji $\lambda \Rightarrow \mu$ a říkáme, že μ lze přímo derivovat z λ v gramatice G (předpokládáme, že je zřejmé, o kterou konkrétní gramatiku se jedná).

Definice 3.6 — derivace Nechť $G = (V, T, P, S)$ je gramatika.

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

   $$u_0 \Rightarrow u_1[p1] \Rightarrow u_2[p_2] \Rightarrow \cdots \Rightarrow u_n[p_n];$$

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

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

   nebo stručněji $u_0 \Rightarrow^n u_n$

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

Definice 3.7 — větná forma, věta Nechť $G = (V, T, P, S)$ je gramatika. Platí-li v G derivace

$$S \Rightarrow^* \alpha$$

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

Definice 3.8 — jazyk generovaný gramatikou Nechť $G = (V, T, P, S)$ je gramatika. *Jazyk* $L(G)$ generovaný touto gramatikou je definován jako

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

3.1.3 Normální formy gramatik

Výše uvedené tvary přepisovacích pravidel u jednotlivých gramatik odpovídají Chomského klasifikaci z roku 1956. Pokud se ale budeme dívat na jednotlivé symboly na levé i pravé straně pravidel, zjistíme, že možných variant a kombinací symbolů z terminální a non-terminální abecedy je velmi mnoho. Různí vědci však v uplynulých desetiletích zjistili, že je možné tvary jednotlivých přepisovacích pravidel bezkontextových, kontextových a neomezených gramatik dále výrazně zjednodušit, aniž by byla snížena jejich generativní síla. Takováto zjednodušení dala vzniknout *normálním formám gramatik*. Ty jsou pro svoji jednoduchost hojně využívány zejména v důkazech, které se tím stávají nepoměrně jednodušší a přehlednější. Uveďme si zde proto pro každý typ gramatik alespoň jednu její normální formu.

Definice 3.9 — Chomského normální forma [30] Bezkontextová gramatika

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

je v *Chomského normální formě*, pokud P obsahuje pouze pravidla tvaru

$$A \rightarrow BC \text{ a } A \rightarrow a,$$

kde $A, B, C \in N$ a $a \in T$. Pokud $\varepsilon \in L(G)$, je přípustné i pravidlo $S \rightarrow \varepsilon$ a S se pak nesmí objevit na pravé straně žádného přepisovacího pravidla.

Věta 3.1 [30] Pro každou bezkontextovou gramatiku $G = (V, T, P, S)$ existuje ekvivalentní bezkontextová gramatika $H = (V_H, T, P_H, S_H)$ v Chomského normální formě taková, že $L(G) = L(H)$.

Definice 3.10 — Greibachové normální forma [13] Bezkontextová gramatika

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

je v *Greibachové normální formě*, pokud P obsahuje pouze pravidla tvaru

$$A \rightarrow aX,$$

kde $A \in N$, $X \in N^*$ a $a \in T$. Pokud $\varepsilon \in L(G)$, je přípustné i pravidlo $S \rightarrow \varepsilon$ a S se pak nesmí objevit na pravé straně žádného přepisovacího pravidla.

Věta 3.2 [13] Pro každou bezkontextovou gramatiku $G = (V, T, P, S)$ existuje ekvivalentní bezkontextová gramatika $H = (V_H, T, P_H, S_H)$ v Greibachové normální formě taková, že $L(G) = L(H)$.

Definice 3.11 — Kurodova normální forma [20] Obecná gramatika

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

je v *Kurodově normální formě*, pokud P obsahuje pouze pravidla tvaru

$$AB \rightarrow CD,\ A \rightarrow BC \text{ a } A \rightarrow a,$$

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

Věta 3.3 [20] Pro každou gramatiku $G = (V, T, P, S)$ existuje gramatika

$$H = (V_H, T, P_H, S_H)$$

v Kurodově normální formě taková, že $L(G) = L(H)$.

Shrňme na závěr této kapitoly o gramatikách, že obecné gramatiky generují třídu rekurzívně vyčíslitelných jazyků (označme ji **RE** z anglického výrazu *recursively enumerable*), kontextové gramatiky generují třídu kontextových jazyků (označme ji **CS** z anglického *context-sensitive*), dále označme **CF** (*context-free*) třídu bezkontextových jazyků generovanou bezkontextovými gramatikami a **REG** třídu regulárních jazyků generovanou regulárními gramatikami.

Věta 3.4 [23] $\mathbf{REG} \subset \mathbf{CF} \subset \mathbf{CS} \subset \mathbf{RE}$

3.2 Automaty a stroje

Zatímco gramatiky jednotlivé věty jazyka generují z daného startovacího symbolu, automaty definují jazyk množinou všech vět, které jsou schopny přijmout a přitom po úplném přečtení vstupního řetězce skončit v některém z cílových stavů. Z tohoto pohledu je můžeme též označit jako akceptory jazyků.

3.2.1 Konečné automaty

Uveďme si nyní formální definici nejjednodušího modelu z oblasti automatů—konečného automatu.

Definice 3.12 — konečný automat *Konečný automat* je pětice

$$M = (Q, \Sigma, R, q_0, F),$$

kde

- Q je konečná množina vnitřních stavů
- Σ je konečná vstupní abeceda
- R je konečná množina pravidel tvaru

  $$pa \to q,$$

  kde $p, q \in Q$ a $a \in \Sigma$; pokud $a \in \Sigma^*$, nazveme takovýto konečný automat *rozšířený*
- $q_0 \in Q$ je počáteční stav
- $F \subseteq Q$ je množina koncových stavů

Z čistě algebraického hlediska je

$$R \subseteq Q\Sigma \times Q$$

$$(\text{příp. } R \subseteq Q\Sigma^* \times Q)$$

konečná binární relace a každé $pa \to q$ je ve skutečnosti uspořádaná dvojice (pa, q), kde $p, q \in Q$ a $a \in \Sigma$ (příp. $a \in \Sigma^*$).

Definice 3.13 — konfigurace Nechť $M = (Q, \Sigma, R, q_0, F)$ je konečný automat. *Konfigurací* automatu M rozumíme řetězec qx, kde $q \in Q$ a $x \in \Sigma^*$. Konfigurací je jednoznačně a úplně popsán momentální celkový stav automatu, kde q představuje aktuální stav a x aktuální obsah vstupní pásky.

Definice 3.14 — přechod Nechť $M = (Q, \Sigma, R, q_0, F)$ je konečný automat a nechť qax a px jsou dvě konfigurace tohoto automatu, $p, q \in Q$, $a \in \Sigma$, $x \in \Sigma^*$. Pokud $qa \to p \in R$, pak automat M provádí *přechod* z konfigurace qax do konfigurace px podle pravidla $qa \to p$ a píšeme

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

nebo stručněji $qax \vdash px$. Symbol $\vdash$ označuje *relaci přechodu.* Analogicky jako v případě definice relace derivace u gramatik, označme postupně $\vdash^n$, $\vdash^+$ a $\vdash^*$ posloupnost přechodů délky n, $n \geq 0$, tranzitivní uzávěr a reflexivní a tranzitivní uzávěr relace přechodu $\vdash$ v tomto pořadí.

Pro naše účely zaveďme v tento okamžik následující notaci. Pokud budou c_1 a c_2 dvě konfigurace a pokud $c_1 \vdash c_2$, budeme v některých případech pro přehlednost používat obrácený zápis pomocí $c_2 \dashv c_1$, kde symbol $\dashv$ (a jeho varianty $\dashv^n$ pro $n \geq 0$, $\dashv^+$ a $\dashv^*$) představuje běžnou relaci $\vdash$ zapsanou obráceně.

Definice 3.15 — jazyk Nechť $M = (Q, \Sigma, R, q_0, F)$ je konečný automat. *Jazyk* přijímaný tímto automatem je definován jako

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

Poznamenejme, že jeden přechod automatu budeme někdy nazývat *výpočetním krokem* a posloupnost těchto kroků *výpočtem*, případně *výpočetním procesem*. Pokud platí $q_0 w \vdash^* f$, nazveme tuto posloupnost *úspěšným výpočtem*.

Označme **FA** třídu jazyků přijímaných konečnými automaty (z jejich anglického názvu *finite automata*).

Věta 3.5 [23] **FA = REG**

3.2.2 Zásobníkové automaty

Zásobníkový automat představuje svým způsobem určité rozšíření konečného automatu v tom smyslu, že obsahuje zásobník jako vnější paměť teoreticky neomezené velikosti. Uveďme si nejprve formální definici.

Definice 3.16 — zásobníkový automat *Zásobníkový automat* je $n-$tice

$$M = (Q, \Sigma_I, \Sigma_{PD}, R, Z, q_0, F),$$

kde

- Q je konečná množina stavů
- Σ_I je konečná vstupní abeceda
- Σ_{PD} je konečná zásobníková abeceda
- R je konečná množina pravidel tvaru

  $$Apa \to wq,$$

  kde $A \in \Sigma_{PD}$, $p, q \in Q$, $w \in \Sigma_{PD}^*$ a $a \in \Sigma_I$; pokud $a \in \Sigma_I^*$, nazveme takovýto zásobníkový automat *rozšířený*
- $Z \in \Sigma_{PD}$ je počáteční symbol zásobníku
- $q_0 \in Q$ je počáteční stav
- $F \subseteq Q$ je množina koncových stavů

Poznamenejme, že z čistě algebraického hlediska je

$$R \subseteq \Sigma_{PD} Q \Sigma \times \Sigma_{PD}^* Q$$

$$(\text{příp. } R \subseteq \Sigma_{PD} Q \Sigma^* \times \Sigma_{PD}^* Q)$$

konečná binární relace a každé pravidlo $Apa \to wq$ je ve skutečnosti uspořádaná dvojice (Apa, wq), kde $p, q \in Q$, $A \in \Sigma_{PD}$, $a \in \Sigma$ (příp. $a \in \Sigma^*$) a $w \in \Sigma_{PD}^*$.

Definice 3.17 — konfigurace Nechť $M = (Q, \Sigma_I, \Sigma_{PD}, R, Z, q_0, F)$ je zásobníkový automat. *Konfigurací* automatu M rozumíme řetězec yqx, kde $y \in \Sigma_{PD}^*$, $q \in Q$ a $x \in \Sigma_I^*$. Konfigurací je kompletně popsán momentální stav automatu, kde y představuje obsah zásobníku, q je aktuální stav a x je momentální obsah vstupní pásky.

Definice 3.18 — přechod Nechť $M = (Q, \Sigma_I, \Sigma_{PD}, R, Z, q_0, F)$ je zásobníkový automat a nechť $uApav$ a $uwqv$ jsou dvě konfigurace tohoto automatu, kde $u \in \Sigma_{PD}^*$, $A \in \Sigma_{PD}$, $a \in \Sigma_I$, $v \in \Sigma_I^*$, $p, q \in Q$ a $w \in \Sigma_{PD}^*$. Pokud $Apa \rightarrow wq \in R$, pak automat M provádí *přechod* z konfigurace $uApav$ do konfigurace $uwqv$ podle pravidla $Apa \rightarrow wq$ a píšeme

$$uApav \vdash uwqv[Apa \rightarrow wq]$$

nebo stručněji $uApav \vdash uwqv$. Podobně jako v případě konečných automatů je symbolem $\vdash$ označena *relace přechodu*. Analogicky jsou definovány i relace $\vdash^n$, $\vdash^+$ a $\vdash^*$, které označují postupně posloupnost přechodů délky n, $n \geq 0$, tranzitivní uzávěr a reflexivní a tranzitivní uzávěr relace přechodu $\vdash$.

Definice 3.19 — jazyk Nechť $M = (Q, \Sigma_I, \Sigma_{PD}, R, Z, q_0, F)$ je zásobníkový automat. *Jazyk* přijímaný automatem M může být definován třemi způsoby.

a) přechodem do koncového stavu:

$$L_f(M) = \{w | w \in \Sigma_I^*, Zq_0w \vdash^* rf, \text{ kde } f \in F \text{ a } r \in \Sigma_{PD}^*\}$$

b) s vyprázdněním zásobníku:

$$L_e(M) = \{w | w \in \Sigma_I^*, Zq_0w \vdash^* \varepsilon p, \text{ kde } p \in Q\}$$

c) přechodem do koncového stavu a s vyprázdněním zásobníku:

$$L_{fe}(M) = \{w | w \in \Sigma_I^*, Zq_0w \vdash^* \varepsilon f, \text{ kde } f \in F\}$$

Podle anglického názvu *pushdown automata* označme **PDA**($Final$) třídu jazyků přijímaných zásobníkovými automaty přechodem do koncového (angl. *final*) stavu, **PDA**($Empty$) třídu jazyků přijímaných zásobníkovými automaty s vyprázdněním (angl. *empty*) zásobníku a **PDA**($Final\&Empty$) třídu jazyků přijímaných zásobníkovými automaty koncovým stavem a s vyprázdněním zásobníku.

Věta 3.6 [23] $\mathbf{PDA}(Final) = \mathbf{PDA}(Empty) = \mathbf{PDA}(Final\&Empty) = \mathbf{CF}$

3.2.3 Turingovy stroje

Turingovy stroje představují nejsilnější výpočetní model, pomocí kterého lze popsat libovolné algoritmizovatelné problémy. Největší význam má zejména v oboru vyčíslitelnosti. Lze jej však použít i pro definici formálních jazyků. Přestože jej přímo nebudeme v našich důkazech využívat, uvedeme si zde pro úplnost jeho formální definici i princip činnosti. Poznamenejme, že se v různé literatuře formální zápis definice Turingova stroje může mírně lišit. My jsme vycházeli z [23].

Definice 3.20 — Turingův stroj *Turingův stroj* je pětice

$$M = (Q, \Sigma, R, s, F),$$

kde

- Q je konečná množina stavů
- Σ je konečná pásková abeceda taková, že $\Sigma \cap Q = \emptyset$ a $\Sigma_I \subseteq \Sigma$, kde Σ_I je vstupní abeceda a $\Sigma - \Sigma_I$ obsahuje $\triangle$ zvaný prázdný symbol
- $R = R_S \cup R_R \cup R_L$ je konečná množina pravidel, kde
  - R_S obsahuje pravidla tvaru
$$qX \to pY \downarrow$$
  - R_R obsahuje pravidla tvaru
$$qX \to pY \hookrightarrow$$
  - R_L obsahuje pravidla tvaru
$$qX \to pY \hookleftarrow$$

  splňující $p, q \in Q$ a $X, Y \in \Sigma$
- $s \in Q$ je počáteční stav
- $F \subseteq Q$ je množina koncových stavů

Poznamenejme, že z čistě algebraického hlediska je

$$R \subseteq Q\Sigma \times Q\Sigma\{\downarrow, \hookleftarrow, \hookrightarrow\}$$

konečná binární relace a každé pravidlo $qX \vdash pYa$ je ve skutečnosti uspořádaná dvojice (qX, pYa), kde $p, q \in Q$, $X, Y \in \Sigma$ a $a \in \{\downarrow, \hookleftarrow, \hookrightarrow\}$.

Definice 3.21 — konfigurace Nechť $M = (Q, \Sigma, R, s, F)$ je Turingův stroj. *Konfigurací* stroje M rozumíme řetězec $xpAy$, kde $x \in \Sigma^*$, $p \in Q$, $A \in \Sigma$ a $y \in \Sigma^*(\Sigma - \{\triangle\}) \cup \{\triangle\}$. Každá konfigurace představuje kompletní popis momentálního celkového stavu automatu a její interpretace je taková, že p je aktuální stav automatu, na pásce je řetězec xAy a A je symbol, na jehož pozici je momentálně umístěna pomyslná čtecí hlava.

Definice 3.22 — přechod Nechť $M = (Q, \Sigma, R, s, F)$ je Turingův stroj a nechť c_1 a c_2 jsou dvě konfigurace tohoto stroje. Říkáme, že M provádí *přechod* z konfigurace c_1 do konfigurace c_2 podle pravidla r a píšeme

$$c_1 \vdash c_2[r]$$

nebo stručněji $c_1 \vdash c_2$, pokud je splněna alespoň jedna z následujících podmínek.

1. *Přechod bez posunu hlavy*
$$c_1 = xpAy, \quad c_2 = xqBy \quad \text{a} \quad r : pA \to qB \downarrow \in R$$
2. *Přechod s posunem hlavy vpravo*

$$c_1 = xpAy, \quad c_2 = xBqy' \quad \text{a} \quad r : pA \to qB \hookrightarrow \in R,$$

kde jestliže $y \neq \varepsilon$, pak $y' = y$ a jestliže $y = \varepsilon$, pak $y' = \triangle$

3. *Přechod s posunem hlavy vlevo*

$$c_1 = xXpAy, \quad c_2 = xqXy' \quad \text{a} \quad r : pA \to qB \hookleftarrow \in R,$$

kde $X \in \Sigma$; jestliže $B \neq \triangle$, pak $y' = By$ a jestliže $B = \triangle$ a $y = \triangle$, pak $y' = \varepsilon$

Podobně jako v případě konečných a zásobníkových automatů je symbolem $\vdash$ označena *relace přechodu*. Analogicky jsou definovány i relace $\vdash^n$, $\vdash^+$ a $\vdash^*$, které označují postupně posloupnost přechodů délky n, $n \geq 0$, tranzitivní uzávěr a reflexivní a tranzitivní uzávěr relace přechodu $\vdash$.

Definice 3.23 — jazyk Nechť $M = (Q, \Sigma, R, s, F)$ je Turingův stroj.

1. M příjímá ε, jestliže
$$s\triangle \vdash^* ufv$$
v M pro nějaké $f \in F$, $u \in \Sigma^*$ a $v \in \Sigma^*(\Sigma - \{\triangle\}) \cup \{\triangle\}$.

2. M přijímá $w \in \Sigma_I^+$, jestliže
$$sw \vdash^* ufv$$
v M pro nějaké $f \in F$, $u \in \Sigma^*$ a $v \in \Sigma^*(\Sigma - \{\triangle\}) \cup \{\triangle\}$.

Jazyk přijímaný Turingovým strojem M je definován jako

$$L(M) = \{w | w \in \Sigma_I^* \text{ a } M \text{ přijímá } w\}.$$

Označíme-li **TM** (z anglického *Turing machines*) třídu všech jazyků přijímaných Turingovými stroji, můžeme shrnout vyjadřovací schopnosti Turingových strojů do následují věty.

Věta 3.7 [23] **TM = RE**

Velmi známou modifikací klasického Turingova stroje je tzv. lineárně ohraničený automat. Lineárně ohraničený automat je Turingův stroj, který nikdy nerozšíří svoji pásku. Přesněji řečeno, pokud obsahuje na své vstupní pásce na počátku řetězec w, nepoužije takový automat více než prvních $|w|$ pozic pásky. K tomuto účelu je zaveden unikátní symbol #, který představuje ukončující symbol a v počáteční konfiguraci obsahuje lineárně ohraničený automat na své pásce řetězec $w\#$. Uveďme si pro úplnost formální definici.

Definice 3.24 — lineárně ohraničený automat *Lineárně ohraničený automat* je pětice

$$M = (Q, \Sigma, R, s, F),$$

kde Q, Σ, R, s a F mají stejný význam jako v případě Turingova stroje (viz Definice 3.20), $\# \in \Sigma_I$ je speciální ukončující symbol a pro každé pravidlo $qX \to pYt \in R$ je buď $\{\#\} \cap \{X, Y\} = \emptyset$, nebo $X = \#$, $Y = \#$ a $t \in \{\downarrow, \hookleftarrow\}$.

Konfigurace, relace přechodu $\vdash$, posloupnost přechodů délky $n \vdash^n$, kde $n \geq 0$, tranzitivní uzávěr $\vdash^+$ a reflexivní a tranzitivní uzávěr $\vdash^*$ jsou definovány stejně jako v případě běžného Turingova stroje.

Jestliže $w\# \vdash^* ufv\#$, kde $w \in (\Sigma_I - \{\#\})^*$, $u, v \in (\Sigma - \{\#\})^*$ a $f \in F$, pak M přijímá w. Jazyk přijímaný lineárně ohraničeným automatem je definován analogicky jako jazyk přijímaný Turingovým strojem, tedy $L(M)$ je množina všech vět, které jsou přijímány lineárně ohraničeným automatem M.

Označíme-li třídu jazyků přijímaných lineárně ohraničenými automaty (z angl. *linear-bounded automata*) jako **LBA**, můžeme vyjadřovací schopnosti těchto automatů popsat formálně následující větou.

Věta 3.8 [23] **LBA = CS**

Klasické konečné a zásobníkové automaty a Turingovy stroje jsou v teorii formálních jazyků všeobecně známé a zejména první dva jmenované modely tvoří významný formální základ pro množství praktických aplikací (hlavně v oblasti lexikální a syntaktické analýzy).

3.3 Další varianty gramatik a automatů

Dosud uvedené gramatiky a automaty představují jakési základní verze, které postupem času vznikly, a jako takové tvoří fundamentální prostředky pro popis formálních jazyků. V současné době je však známo mnoho dalších formálních modelů pro popis jazyků a stále se objevují další modely, které využívají některý běžný automat nebo gramatiku jako základ, zavádí ale nové prvky, modifikace, případně omezení a tím mění vyjadřovací sílu původního modelu. To je v drtivé většině těchto případů cílem—využít nějaký jednodušší model a vhodnou modifikací zvýšit jeho schopnosti.

I my se v této práci o něco podobného budeme pokoušet. Zmiňme se proto o některých dalších formálních modelech používaných pro popis jazyků. Zdůrazněme však, že se zdaleka nejedná o úplný přehled. Různých takových formálních modelů je velmi mnoho a jejich kompletní výčet by byl neúměrně dlouhý.

Uveďme však alespoň některé. Formálně popíšeme ty modely, které budeme dále využívat. Konkrétně se jedná o frontové gramatiky a jejich modifikovanou variantu—levě rozšířené frontové gramatiky.

3.3.1 Frontové gramatiky

Jednou z gramatik, která se svým tvarem pravidel a způsobem generování řetězců výrazně liší od běžných gramatik Chomského hierarchie, je frontová gramatika. Její síla je však stejná jako síla obecných gramatik. Z frontových gramatik budeme vycházet při studiu oboustranných zásobníkových automatů a proto si zde jejich definici uvedeme formálně.

Definice 3.25 — frontová gramatika *Frontová gramatika* je šestice

$$Q = (V, T, W, F, s, P),$$

kde

- V je konečná množina symbolů, W je konečná množina stavových symbolů a platí $V \cap W = \emptyset$
- $T \subset V$ je množina terminálních symbolů
- $F \subseteq W$ je množina koncových stavových symbolů
- $s \in (V - T)(W - F)$ je výchozí axiom
- $P \subseteq V \times (W - F) \times V^* \times W$ je konečná relace taková, že pro každé $a \in V$ existuje nějaký prvek $(a, b, x, c) \in P$; prvky této množiny nazýváme přepisovací pravidla, nebo krátce pravidla

Definice 3.26 — přímá derivace, derivace Nechť $Q = (V, T, W, F, s, P)$ je frontová gramatika. 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

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

v Q, nebo stručněji $arb \Rightarrow rxc$. Symbolem $\Rightarrow$ je podobně jako v případě obecných gramatik označována *relace přímé derivace* v Q. Obvyklým způsobem jsou pak definovány i relace $\Rightarrow^n$, $n \geq 0$, $\Rightarrow^+$ a $\Rightarrow^*$.

Protože frontové gramatiky nejsou příliš rozšířené a jejich princip činnosti se může na první pohled jevit jako složitý, uveďme si na tomto místě podrobnější vysvětlení výše zapsaných formalismů.

Výchozí axiom je vždy tvaru $s = ap$, kde $a \in (V - T)$ a $p \in (W - F)$. Položme $N = V - T$ a všimněme si podobnosti symbolů s běžnými gramatikami. Množinu V můžeme i zde chápat jako úplnou abecedu. N resp. T je pak abeceda nonterminálních resp. terminálních symbolů. Z terminálních symbolů jsou složeny jednotlivé věty generovaného jazyka. Až dosud zde není v případě těchno abeced žádná odlišnost od běžných gramatik.

Frontová gramatika však obsahuje navíc další množinu W tzv. stavových symbolů. Její podmnožinou je množina F s tzv. koncovými stavovými symboly. W a F tedy můžeme chápat podobně jako množinu stavů a koncových stavů u automatů. Frontové gramatiky tedy využívají jak nonterminální a terminální symboly, tak i stavové symboly, které se podílejí na řízení derivačního procesu.

Pojďme si nyní podrobně rozebrat přímý derivační krok popsaný v Definici 3.26. Řetězce $u = arb$ a $v = rxc$ z této definice můžeme nazvat větnými formami a čtveřici $(a, b, x, c) \in P$ je již zmíněné přepisovací pravidlo. Ve větné formě arb je $a \in V$ jejím nejlevějším symbolem, $r \in V^*$ je zbytek dosud vygenerovaného řetězce tvořeného terminálními a nonterminálními symboly (bez prvního symbolu a) a konečně $b \in W$ je aktuální stavový symbol pro tuto větnou formu. Přímá derivace

$$arb \rightarrow rxc$$

pomocí pravidla $(a, b, x, c) \in P$ je provedena následovně.

Klíčovými symboly pro aplikaci tohoto pravidla je první symbol a větné formy a aktuální stav b přiřazený k této větné formě. Tyto symboly musí být shodné s prvním a druhým symbolem přepisovacího pravidla (v uvedeném pořadí). Tímto je ve skutečnosti vybráno jedno z přepisovacích pravidel, které bude použito. Derivační krok je pak složen z těchto akcí:

1. odebrání prvního symbolu a z aktuální větné formy

2. přidání řetězce x z přepisovacího pravidla na konec řetězce r v aktuální větné formě

3. změna aktuálního stavového symbolu z b na c a jeho přidání na samý konec dosud vygenerovaného řetězce rx, čímž získáme novou větnou formu

Tím je derivační krok kompletní. Z jeho struktury je pak již zřejmý název gramatik tohoto typu, tedy frontové. Všimněme si, že přepisovací pravidla pracují (kromě stavových symbolů) s větnou formou jako s frontou, kdy vždy odeberou z jejího čela jeden symbol a na její konec vloží symboly další (ve formě řetězce).

Výchozí axiom $s = ap$ potom představuje minimální strukturu, ze které je možné zahájit derivaci. Fronta v tomto případě obsahuje jediný symbol a. Stavový symbol p pak lze chápat podobně jako počáteční stav u automatů. Derivace může skončit dvěma způsoby.

a) není aplikovatelné žádné přepisovací pravidlo; přitom však větná forma obsahuje nonterminální symboly, nebo nebylo dosaženo některého z koncových stavových symbolů

b) větná forma neobsahuje žádné nonterminální symboly a bylo dosaženo koncového stavového symbolu; v tomto případně byla derivace úspěšná a vygenerovaný řetězec před stavovým symbolem představuje platnou větu jazyka

Nyní si již můžeme formálně zapsat jazyk generovaný frontovou gramatikou.

Definice 3.27 — jazyk Nechť $Q = (V, T, W, F, s, P)$ je frontová gramatika. *Jazyk* $L(Q)$ generovaný touto gramatikou je definován jako

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

Př.: Protože frontové gramatiky nejsou v teorii formálních jazyků příliš rozšířené, uveďme si pro ilustraci příklad, aby bylo zřejmé, jakým způsobem tyto gramatiky generují jednotlivé věty jazyka. Uvažujme frontovou gramatiku

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

kde jednotlivá pravidla gramatiky jsou uvedena v následující tabulce.

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

Jednou z derivací v této gramatice je

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

která generuje větu $abb \in L(Q)$.

Z [22] pro nás bude užitečná jedna z modifikací frontové gramatiky, tzv. levě rozšířená frontová gramatika.

Definice 3.28 — levě rozšířená frontová gramatika *Levě 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 běžné frontové gramatiky a množina pravidel P je definována jako konečná relace

$$P \subseteq V \times (W - F) \times V^* \times W.$$

Tato definice na rozdíl od frontové gramatiky striktně nevyžaduje, aby pro každé $a \in V$ existoval nějaký prvek $(a, b, x, c) \in P$.

Definice 3.29 — přímá derivace, derivace Nechť

$$Q = (V, T, W, F, s, P)$$

je levě rozšířená frontová gramatika a nechť $\# \notin V \cup W$. 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

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

v Q, nebo stručněji $w\#arb \Rightarrow wa\#rxc$, kde $\Rightarrow$ opět označuje relaci přímé derivace. Standardním způsobem jsou pak také definovány relace $\Rightarrow^n$, kde $n \geq 0$, $\Rightarrow^+$ a $\Rightarrow^*$.

Definice 3.30 — jazyk Nechť $Q = (V, T, W, F, s, P)$ je levě rozšířená frontová gramatika. *Jazyk* generovaný levě rozšířenou frontovou gramatikou je definován jako

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

Všimněme si, že levě rozšířena frontová gramatika pracuje stejně jako běžná frontová gramatika s tím rozdílem, že vlevo od oddělovacího symbolu # postupně zaznamenává symboly z V z první složky použitých přepisovacích pravidel.

Př.: Pro ilustraci uvažujme stejnou gramatiku jako v předchozím příkladu. Rovněž napíšeme derivační posloupnost stejné věty, aby byla vidět zásadní odlišnost levě rozšířené gramatiky od frontové gramatiky. Derivace věty abb $\in L(G)$ v levě rozšířené frontové gramatice vypadá následovně.

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

Jak jsme si již uvedli v definici, levě rozšířená frontová gramatika zaznamenává symboly z prvních komponent použitých pravidel. V poslední větné formě můžeme tak najít vlevo od symbolu # jejich kompletní posloupnost $AAABB$. Všimněme si v této souvislosti použitých přepisovacích pravidel a jejich prvních komponent—skutečně tvoří uvedenou posloupnost. Této vlastnosti s výhodou využijeme při studiu oboustranných zásobníkových automatů.

Označme **QG** (z angl. *queue grammars*) třídu jazyků generovaných frontovými gramatikami a **LEQG** (z angl. *left-extended queue grammars*) třídu jazyků generovaných levě rozšířenými frontovými gramatikami.

Věta 3.9 [17], [22] **RE = QG = LEQG**

Zmiňme se na závěr této kapitoly alespoň slovně o některých dalších formálních modelech. V [8] se autor podrobně zabývá tzv. bezkontextovými gramatikami nad volnými grupami [2], [3]. Jak již název napovídá, základem je běžná bezkontextová gramatika. Relace přímé derivace však není definována nad volným monoidem, ale nad volnou grupou generovanou totální abecedou dané gramatiky. V naší práci tyto gramatiky navíc vylepšíme tak, že zredukujeme nutný počet nonterminálních symbolů. Podobný přístup je rovněž paralelizovatelný prostřednictvím E0L gramatik nad volnými grupami [8], [7].

Bezkontextové gramatiky jsou základem mnoha dalších struktur pro popis jazyků. Jmenujme například maticové a programované gramatiky, gramatiky s náhodným kontextem, atd. [27]. Tyto gramatiky zavádějí tzv. řízené přepisování [10], které omezuje možnost použití některých přepisovacích pravidel v daném kroku a do jisté míry řídí derivační proces jako celek. Tvary bezkontextových pravidel (s možností výskytu terminálního symbolu na levé straně) využívají i gramatiky z rodiny 0L-systémů [27], případně gramatiky s kontextovými podmínkami [24]. Rozsáhlou a stále intenzivně zkoumanou skupinou jsou gramatické systémy. Gramatický systém je tvořen několika bezkontextovými gramatikami, tzv. komponentami, které v průběhu derivačního procesu do určité míry spolupracují a jako celek jsou silnější než jediná bezkontextová gramatika. Dostupných publikací zabývajících se touto problematikou je v současné době celá řada.

Z oblasti automatů uveďme například $k-$obrátkové zásobníkové automaty, které zavádějí pojem tzv. obrátky zásobníku. Takovéto omezení však sílu zásobníkového automatu nezvýší. Jazyky přijímané $k-$obrátkovými zásobníkovými automaty tvoří nekonečnou hierarchii, kdy se pro neomezené k dostáváme k běžnému zásobníkovému automatu.

Další modifikací zásobníkových automatů jsou automaty dvou a vícezásobníkové [23]. Je však dokázáno [23], že zvyšování počtu zásobníků na tři a více již nijak sílu automatu nezvýší. Speciální variantou dvouzásobníkových automatů jsou souběžně jednoobrátkové dvouzásobníkové automaty [22], které přijímají každou větu jazyka s jedinou souběžnou obrátkou obou zásobníků. Těmito automaty jsme se inspirovali pro další výzkum, ve kterém definujeme a dále studujeme oboustranné zásobníkové automaty. Jejich vlastnosti a výhody oproti souběžně jednoobrátkovým dvouzásobníkovým automatům (a vůbec oproti vícezásobníkovým automatům) popíšeme v další kapitole.

V [29] a [15] autoři prezentují velmi jednoduchou modifikaci zásobníkových automatů, kterým přidali možnost reverzovat obsah zásobníku v průběhu výpočetního procesu. Tím vytvořili formální model, který s ohledem na počet těchto reverzací generuje nekonečnou hierarchii jazyků.

Uplatnění již dříve zmiňovaných volných grup našli v teorii automatů i autoři [9], kde jsou definovány konečné automaty nad volnými grupami. Dále jmenujme např. frontové automaty [6], které místo zásobníků zavádějí frontu. Další modifikace zavádí pro změnu určitý stupeň řízení zásobníkových automatů pomocí regulárního, případně lineárního jazyka [18].

Tato kapitola měla kromě definice frontových gramatik poskytnout i jakýsi náhled na některé již známé modifikace základních formálních modelů. Nyní se tedy dostáváme k hlavním výsledkům této práce. Jak jsme se již zmínili v úvodu, zavedeme některé modifikace do již známých formálních modelů pro popis jazyků a budeme studovat vliv těchto modifikací na jejich vyjadřovací sílu. Jako první budeme modifikovat běžné zásobníkové automaty. Dále představíme tzv. vertikální kontext v obecných gramatikách. Jako poslední výsledek představíme bezkontextové gramatiky nad volnými grupami s redukovaným počtem nonterminálů. Přestože je společným sjednocujícím hlediskem zavádění těchto modifikací do známých formálních modelů, budeme prezentovat jednotlivé výsledky v samostatných kapitolách, neboť podstatou činnosti se oba studované modely výrazně liší.

Kapitola 4

Oboustranné zásobníkové automaty

V této a v následujících dvou kapitolách již budeme prezentovat nové výsledky našeho výzkumu.

Jak jsme se zmínili na počátku, pokusíme se vytvořit další modifikaci běžného zásobníkového automatu—tzv. oboustranný zásobníkový automat [4]—a budeme studovat schopnosti nově vzniklé struktury. Hlavní inspirací pro tento směr výzkumu byly souběžně jednoobrátkové dvouzásobníkové automaty [22]. Jak je v [22] 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.

Následující pomocná věta bude využita při konstrukci oboustranného zásobníkového automatu.

Lemma 4.1 [**22**] 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 #.

4.1 Základní definice

Uveďme si nyní formální definici oboustranného zásobníkového automatu.

Definice 4.1 — 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

  $$Z_1|Z_2px \to \gamma_1|\gamma_2q$$

  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

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

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

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

Definice 4.2 — 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 4.3 — přechod Nechť

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

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

$$L\gamma Rpxy \vdash \gamma_L \gamma \gamma_R qy[L|Rpx \to \gamma_L|\gamma_R q]$$

nebo stručněji $L\gamma Rpxy \vdash \gamma_L \gamma \gamma_R qy$. 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 4.2, představuje každá konfigurace úplný popis celkového stavu automatu v jednom okamžiku. Počáteční konfigurací je s ohledem na definici konfigurace $Z_L Z_R zw$, 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.

Popišme si zde neformálně jeden výpočetní krok (přechod) automatu. Uvažujme dvě konfigurace $L\gamma Rpxy$ a $\gamma_L \gamma \gamma_R qy$ z předchozí definice a pravidlo $L|Rpx \to \gamma_L|\gamma_R q \in R$. V první konfiguraci představuje řetězec $L\gamma R$ aktuální obsah zásobníku, kde na jeho levém vrcholu je symbol L, na jeho pravém vrcholu pak je symbol R. Aktuálním stavem automatu je stav p a na vstupní pásce je řetězec xy, kde x je jeho nejlevější symbol (na jehož pozici se nachází pomyslná čtecí hlava). Poznamenejme, že v případě rozšířeného oboustranného zásobníkového automatu může být $x \in T^*$ a čtecí hlava pak může najednou načíst celý řetězec x. Aplikaci pravidla $L|Rpx \to \gamma_L|\gamma_R$ na právě popsanou větnou formu lze popsat následujícími body.

1. odstranění levého a pravého vrcholu zásobníku (L a R)
2. přečtení x ze vstupní pásky
3. vložení řetězce γ_L zleva na zásobník a řetězce γ_R zprava na zásobník (místo odebraných symbolů z obou vrcholů)
4. změna aktuálního stavu automatu z p na q

Tím je jeden výpočetní krok automatu kompletní. Pokud je možné sekvencí kroků provedených z počáteční konfigurace zcela načíst vstupní řetězec na pásce, dosáhnout některého z koncových stavů z množiny F_M a zcela vyprázdnit zásobník, říkáme, že byl vstupní řetězec přijat. V opačném případě by byl odmítnut. Jazyk definovaný oboustranným zásobníkovým automatem je pak možné formalizovat následovně.

Definice 4.4 — jazyk *Jazyk* přijímaný oboustranným zásobníkovým automatem M je definován jako

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

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 4.5 — 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.

4.2 Vyjadřovací síla

Nyní se již dostáváme k hlavním výsledkům této kapitoly. Dokážeme, že oboustranné zásobníkové automaty jsou schopny generovat celou třídu rekurzívně vyčíslitelných jazyků. Kromě toho je možné výrazně redukovat počet symbolů zásobníkové abecedy. Obě verze automatu navíc přijímají každou větu jazyka s jedinou obrátkou zásobníku. 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.

Věta 4.2 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 $Z_L|Z_Rz \to Z_L|SZ_R\langle q_0, 1\rangle$ 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 $b = 1, \ldots, 5$. 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.: Ještě než přistoupíme k rigoróznímu důkazu, uveďme si pro ozřejmení principu činnosti automatu praktický příklad, na kterém automat zkonstruujeme a ukážeme, jakým způsobem přijímá vstupní řetězec. Budeme vycházet ze stejné frontové gramatiky, kterou jsme použili v předchozích příkladech. Jedná se o 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:

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

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
$1a)$	$Z_L\|Z_R z$	$\rightarrow$	$Z_L\|AZ_R\langle p,1\rangle$	—
$2a)$	$Z_L\|Z_R\langle p,1\rangle$	$\rightarrow$	$Z_L\overline{A}\|AAZ_R\langle p,1\rangle$	(A,p,AA,p)
$2b)$	$Z_L\|Z_R\langle p,1\rangle$	$\rightarrow$	$Z_L\overline{A}\|BBZ_R\langle p,1\rangle$	(A,p,BB,p)
$3a)$	$Z_L\|Z_R\langle p,1\rangle$	$\rightarrow$	$Z_L\|Z_R\langle p,2\rangle$	—
$3b)$	$Z_L\|Z_R\langle q,1\rangle$	$\rightarrow$	$Z_L\|Z_R\langle q,2\rangle$	—
$3c)$	$Z_L\|Z_R\langle f,1\rangle$	$\rightarrow$	$Z_L\|Z_R\langle f,2\rangle$	—
$4a)$	$Z_L\|Z_R\langle p,2\rangle a$	$\rightarrow$	$Z_L\overline{A}\|Z_R\langle q,2\rangle$	(A,p,a,q)
$4b)$	$Z_L\|Z_R\langle q,2\rangle a$	$\rightarrow$	$Z_L\overline{A}\|Z_R\langle q,2\rangle$	(A,q,a,q)
$4c)$	$Z_L\|Z_R\langle q,2\rangle b$	$\rightarrow$	$Z_L\overline{B}\|Z_R\langle q,2\rangle$	(B,q,b,q)
$5a)$	$Z_L\|Z_R\langle q,2\rangle b$	$\rightarrow$	$\overline{B}\|\varepsilon f$	(B,q,b,f)
$6a)$	$\overline{A}\|Af$	$\rightarrow$	$\varepsilon\|\varepsilon f$	—
$6b)$	$\overline{B}\|Bf$	$\rightarrow$	$\varepsilon\|\varepsilon f$	—

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.

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

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.

Rigorózní důkaz

Postupnou demonstrací platnosti tvrzení týkajících se jednotlivých etap činnosti oboustranného zásobníkového automatu a levě rozšířené frontové gramatiky dokážeme inkluze $L(G) \subseteq L(M)$ a $L(M) \subseteq L(G)$.

Tvrzení A. Jestliže

$$A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^i A_1 \dots A_n B_1 \dots B_i \# B_{i+1} \dots B_m x_1 \dots x_i p$$

v G, potom

$$
\begin{array}{ll}
 & Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle u,1\rangle \omega \\
\vdash^i & Z_L \overline{B_i} \dots \overline{B_1 A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m x_1 \dots x_i Z_R \langle p,1\rangle \omega
\end{array}
$$

v M, kde $A_1, \dots, A_n, B_1, \dots, B_m \in (V-T)$, $x_1, \dots, x_i \in (V-T)^*$, $u, p \in (W-F)$, $m \geq 0$, $n \geq 0$, $\omega \in T^*$, $i < m$.

Základ indukce. Nechť $i = 0$. Pak

$$A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^0 A_1 \dots A_n \# B_1 \dots B_m u$$

v G. Je zřejmé, že i

$$Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle u,1\rangle \omega \vdash^0 Z_L \overline{A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle u,1\rangle \omega$$

v M.

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

Indukční krok. Uvažujme v G libovolnou derivaci tvaru

$$A_1 \ldots A_n \# B_1 \ldots B_m u \Rightarrow^{l+1} A_1 \ldots A_n B_1 \ldots B_l B_{l+1} \# B_{l+2} \ldots B_m x_1 \ldots x_l x_{l+1} q$$

a vyjádřeme ji jako

$$\begin{array}{ll} & A_1 \ldots A_n \# B_1 \ldots B_m u \\ \Rightarrow^l & A_1 \ldots A_n B_1 \ldots B_l \# B_{l+1} \ldots B_m x_1 \ldots x_l p \\ \Rightarrow & A_1 \ldots A_n B_1 \ldots B_l B_{l+1} \# B_{l+2} \ldots B_m x_1 \ldots x_l x_{l+1} q, \end{array}$$

kde $l + 2 \leq m$ a $q \in (W - F)$.

Podle indukční hypotézy

$$\begin{array}{ll} & Z_L \overline{A_n} \ldots \overline{A_1} A_1 \ldots A_n B_1 \ldots B_m Z_R \langle u, 1 \rangle \omega \\ \vdash^l & Z_L \overline{B_l} \ldots \overline{B_1 A_n} \ldots \overline{A_1} A_1 \ldots A_n B_1 \ldots B_m x_1 \ldots x_l Z_R \langle p, 1 \rangle \omega \\ \vdash & Z_L \overline{B_{l+1} B_l} \ldots \overline{B_1 A_n} \ldots \overline{A_1} A_1 \ldots A_n B_1 \ldots B_m x_1 \ldots x_l x_{l+1} Z_R \langle q, 1 \rangle \omega \end{array}$$

v M. V P existuje pouze jeden typ pravidel schopných provést derivaci

$$\begin{array}{ll} & A_1 \ldots A_n B_1 \ldots B_l \# B_{l+1} \ldots B_m x_1 \ldots x_l p \\ \Rightarrow & A_1 \ldots A_n B_1 \ldots B_l B_{l+1} \# B_{l+2} \ldots B_m x_1 \ldots x_l x_{l+1} q \end{array}$$

v G a to pravidla tvaru (B_{l+1}, p, x_{l+1}, q), kde $B_{l+1} \in (V - T)$, $p, q \in (W - F)$ a $x_{l+1} \in (V - T)^*$. Všimněme si, že podle bodu 2 konstrukce existuje v R pravidlo

$$Z_L | Z_R \langle p, 1 \rangle \rightarrow Z_L \overline{B_{l+1}} | x_{l+1} Z_R \langle q, 1 \rangle$$

takže

$$\begin{array}{ll} & Z_L \overline{B_l} \ldots \overline{B_1 A_n} \ldots \overline{A_1} A_1 \ldots A_n B_1 \ldots B_m x_1 \ldots x_l Z_R \langle p, 1 \rangle \omega \\ \vdash & Z_L \overline{B_{l+1} B_l} \ldots \overline{B_1 A_n} \ldots \overline{A_1} A_1 \ldots A_n B_1 \ldots B_m x_1 \ldots x_l x_{l+1} Z_R \langle q, 1 \rangle \omega \end{array}$$

v M a Tvrzení A tedy platí. □

Tvrzení B. Jestliže

$$\begin{array}{ll} & A_1 \ldots A_n \# B_1 \ldots B_m a_1 \ldots a_k u \\ \Rightarrow^i & A_1 \ldots A_n B_1 \ldots B_i \# B_{i+1} \ldots B_m a_1 \ldots a_k b_1 \ldots b_i p \end{array}$$

v G, pak

$$\begin{array}{ll} & Z_L \overline{A_n} \ldots \overline{A_1} A_1 \ldots A_n B_1 \ldots B_m Z_R \langle u, 2 \rangle b_1 \ldots b_j \\ \vdash^i & Z_L \overline{B_i} \ldots \overline{B_1 A_n} \ldots \overline{A_1} A_1 \ldots A_n B_1 \ldots B_m Z_R \langle p, 2 \rangle b_{i+1} \ldots b_j \end{array}$$

v M, kde $A_1, \ldots, A_n, B_1, \ldots, B_m \in (V - T)$, $a_1, \ldots, a_k, b_1, \ldots, b_j \in T^*$, $u, p \in (W - F)$, $i < m$, $i < j$, $k \geq 0$ a $j \geq 0$.

Základ indukce. Nechť $i = 0$. Potom

$$A_1 \ldots A_n \# B_1 \ldots B_m a_1 \ldots a_k u \Rightarrow^0 A_1 \ldots A_n \# B_1 \ldots B_m a_1 \ldots a_k u$$

v G. Rozhodně také

$$\begin{array}{ll} & Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j \\ \vdash^0 & Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j \end{array}$$

v M.

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

Indukční krok. Uvažujme v G libovolnou derivaci tvaru

$$\begin{array}{ll} & A_1\dots A_n\#B_1\dots B_ma_1\dots a_ku \\ \Rightarrow^{l+1} & A_1\dots A_nB_1\dots B_lB_{l+1}\#B_{l+2}\dots B_ma_1\dots a_kb_1\dots b_lb_{l+1}q \end{array}$$

a vyjádřeme tuto derivaci jako

$$\begin{array}{ll} & A_1\dots A_n\#B_1\dots B_ma_1\dots a_ku \\ \Rightarrow^{l} & A_1\dots A_nB_1\dots B_l\#B_{l+1}\dots B_ma_1\dots a_kb_1\dots b_lp \\ \Rightarrow & A_1\dots A_nB_1\dots B_lB_{l+1}\#B_{l+2}\dots B_ma_1\dots a_kb_1\dots b_lb_{l+1}q, \end{array}$$

kde $l+2 \leq m$, $k \geq 0$, $q \in (W-F)$.

Podle indukční hypotézy

$$\begin{array}{ll} & Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j \\ \vdash^l & Z_L\overline{B_l}\dots\overline{B_1A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle p,2\rangle b_{l+1}\dots b_j \\ \vdash & Z_L\overline{B_{l+1}B_l}\dots\overline{B_1A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle q,2\rangle b_{l+2}\dots b_j \end{array}$$

v M.

V tomto případě existuje pouze jedna možnost jak může G provést derivaci

$$\begin{array}{ll} & A_1\dots A_nB_1\dots B_l\#B_{l+1}\dots B_ma_1\dots a_kb_1\dots b_lp \\ \Rightarrow & A_1\dots A_nB_1\dots B_lB_{l+1}\#B_{l+2}\dots B_ma_1\dots a_kb_1\dots b_lb_{l+1}q \end{array}$$

a to použitím některého pravidla tvaru $(B_{l+1}, p, b_{l+1}, q) \in P$, kde $B_{l+1} \in (V-T)$, $p, q \in (W-F)$, $b_{l+1} \in T^*$. Podle bodu 4 konstrukce existuje v R pravidlo tvaru

$$Z_L|Z_R\langle p,2\rangle b_{l+1} \rightarrow Z_L\overline{B_{l+1}}|Z_R\langle q,2\rangle$$

takže

$$\begin{array}{ll} & Z_L\overline{B_l}\dots\overline{B_1A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle p,2\rangle b_{l+1}\dots b_j \\ \vdash & Z_L\overline{B_{l+1}B_l}\dots\overline{B_1A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle q,2\rangle b_{l+2}\dots b_j \end{array}$$

v M a Tvrzení B platí. □

Tvrzení C. Jestliže

$$A_1\dots A_{n-1}\#A_nyq \Rightarrow A_1\dots A_{n-1}A_n\#yzt$$

v G, kde $A_1,\dots,A_n \in (V-T)$, $y, z \in T^*$, $q \in (W-F)$, $t \in F$, pak

$$Z_L\overline{A_{n-1}}\dots\overline{A_1}A_1\dots A_nZ_R\langle q,2\rangle z \vdash \overline{A_n}\dots\overline{A_1}A_1\dots A_nf\varepsilon$$

v M, kde $f \in F_M$.

Gramatika G provádí tuto derivaci pomocí některého pravidla tvaru $(A_n, q, z, t) \in P$, kde $A_n \in (V - T)$, $z \in T^*$, $q \in (W - F)$, $t \in F$. Podle bodu 5 konstrukce obsahuje R pravidlo

$$Z_L|Z_R\langle q, 2\rangle z \to \overline{A_n}|\varepsilon f,$$

takže

$$Z_L\overline{A_{n-1}}\dots\overline{A_1}A_1\dots A_nZ_R\langle q, 2\rangle z \vdash \overline{A_n}\dots\overline{A_1}A_1\dots A_nf\varepsilon$$

v M a Tvrzení C tedy platí. □

Tvrzení A, B a C dohromady potvrzují platnost inkluze $L(G) \subseteq L(M)$. Tím je první část důkazu hotova. Ve druhé části budeme demonstrovat platnost tvrzení kterými dokážeme inkluzi $L(M) \subseteq L(G)$.

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

$Z_LZ_Rzw_1w_2\dots w_r \vdash$
$Z_LSZ_R\langle q_0, 1\rangle w_1w_2\dots w_r \vdash$
$Z_L\overline{S}SX^1_1X^1_2\dots X^1_{n_1}Z_R\langle q_1, 1\rangle w_1w_2\dots w_r \vdash$
$Z_L\overline{X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots X^2_{n_2}Z_R\langle q_2, 1\rangle w_1w_2\dots w_r \vdash$
$Z_L\overline{X^1_2}\overline{X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots X^2_{n_2}X^3_1X^3_2\dots X^3_{n_3}Z_R\langle q_3, 1\rangle w_1w_2\dots w_r \vdash$

$\dots$

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

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

$Z_L\overline{X^k_{j+1}X^k_j}\dots\overline{X^1_2}\overline{X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots X^2_{n_2}X^3_1X^3_2\dots X^3_{n_3}\dots$
$\dots X^m_1X^m_2\dots X^m_{n_m}Z_R\langle q_{m+1}, 2\rangle w_2\dots w_r \vdash$

$Z_L\overline{X^k_{j+2}X^k_{j+1}X^k_j}\dots\overline{X^1_2}\overline{X^1_1}\overline{S}SX^1_1X^1_2\dots X^1_{n_1}X^2_1X^2_2\dots X^2_{n_2}X^3_1X^3_2\dots X^3_{n_3}\dots$
$\dots X^m_1X^m_2\dots X^m_{n_m}Z_R\langle q_{m+2}, 2\rangle w_3\dots w_r \vdash$

$\dots$

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

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

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

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

$\ldots$

$\overline{X^1_1}\overline{S}SX^1_1f\varepsilon\vdash$
$\overline{S}Sf\varepsilon\vdash$
$\varepsilon f\varepsilon$

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

Postupně projdeme všechny kroky konstrukce množiny pravidel R. Zopakujme, že v každé úspěšné výpočetní sekvenci (tedy kdy je vstupní řetězec přijat), používá M pravidla zkonstruovaná v kroku b před tím, než použije pravidla zkonstruovaná v kroku $b+1$, pro $b = 1,\ldots,5$.

V prvním kroku automat M použije pravidlo

$$Z_L|Z_Rz\rightarrow Z_L|SZ_R\langle q_0,1\rangle$$

zkonstruované v bodu 1, kde Sq_0 je výchozí axiom gramatiky G. To je jediný způsob, kterým může M provést přechod

$$Z_LZ_Rzw_1w_2\ldots w_r\vdash Z_LSZ_R\langle q_0,1\rangle w_1w_2\ldots w_r.$$

Všimněme si, že toto pravidlo je použito právě jednou v průběhu celé úspěšné výpočetní sekvence automatu. Tímto krokem M přepne do módu generování nonterminálů.

V posloupnosti přechodů

$Z_LSZ_R\langle q_0,1\rangle w_1w_2\ldots w_r\vdash^*$
$Z_L\overline{X^k_j}\ldots\overline{X^1_2X^1_1}\overline{S}SX^1_1X^1_2\ldots X^1_{n_1}X^2_1X^2_2\ldots X^2_{n_2}X^3_1X^3_2\ldots X^3_{n_3}\ldots$
$\ldots X^m_1X^m_2\ldots X^m_{n_m}Z_R\langle q_m,1\rangle w_1w_2\ldots w_r$

M používá pravidla tvaru

$$Z_L|Z_R\langle q,1\rangle\rightarrow Z_L\overline{A}|xZ_R\langle p,1\rangle$$

zkonstruovaná v bodu 2, kde $A\in(V-T)$, $x\in(V-T)^*$, $p,q\in(W-F)$. Tato část výpočtu je charakterizována stavy automatu tvaru $\langle q,1\rangle$. Detailní důkaz této části je popsán Tvrzením E.

Přechodem

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

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

je M přepnut do módu čtení terminálů. K přepnutí slouží pravidlo tvaru

$$Z_L|Z_R\langle q,1\rangle \rightarrow Z_L|Z_R\langle q,2\rangle$$

zkonstruované v bodu 3. Poznamenejme, že toto pravidlo je použito vždy právě jednou v průběhu libovolné úspěšné výpočetní sekvence. Protože toto pravidlo zároveň změní aktuální stav automatu tvaru $\langle q,1\rangle$ na stav tvaru $\langle q,2\rangle$, kde $q \in (W-F)$, není po provedení tohoto kroku žádná další možnost pro použití pravidel z konstruovaných v bodech 1 až 3.

Dále následuje posloupnost přechodů

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

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

ve které M aplikuje pravidla zkonstruovaná v bodu 4 a čte vstupní řetězec tvořených terminálními symboly. Podrobný důkaz této části je popsán Tvrzením F.

V kroku

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

$$\overline{X_{n_m}^mX_{n_m-1}^m}\dots\dots\overline{X_{j+2}^kX_{j+1}^kX_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots$$
$$\dots X_1^mX_2^m\dots X_{n_m}^mf\varepsilon \vdash$$

se M dostane do koncového stavu. To je zajištěno pravidlem tvaru

$$Z_L|Z_R\langle q,2\rangle y \rightarrow \overline{A}|\varepsilon f$$

zkonstruovaným v bodu 5, kde $q \in (W-F)$, $y \in T^*$, $A \in (V-T)$ and $f \in F_M$.

V tuto chvíli zbývá splnit poslední podmínku pro přijetí řetězce—vyprázdnit zásobník. To je vyjádřeno sekvencí přechodů

$$\overline{X_{n_m}^mX_{n_m-1}^m}\dots\dots\overline{X_{j+2}^kX_{j+1}^kX_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots$$
$$\dots X_1^mX_2^m\dots X_{n_m}^mf\varepsilon \vdash$$

$$\overline{X_{n_m-1}^m}\dots\dots\overline{X_{j+2}^kX_{j+1}^kX_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots$$
$$\dots X_1^mX_2^m\dots X_{n_m-1}^mf\varepsilon \vdash$$

$$\overline{X_{n_m-2}^m}\dots\dots\overline{X_{j+2}^kX_{j+1}^kX_j^k}\dots\overline{X_2^1X_1^1}\overline{S}SX_1^1X_2^1\dots X_{n_1}^1X_1^2X_2^2\dots X_{n_2}^2X_1^3X_2^3\dots X_{n_3}^3\dots$$
$$\dots X_1^mX_2^m\dots X_{n_m-2}^mf\varepsilon \vdash$$

$\dots$

$\overline{X_1^1}\overline{S}SX_1^1f\varepsilon \vdash$
$\overline{S}Sf\varepsilon \vdash$
$\varepsilon f\varepsilon.$

Zásobník je systematicky vyprazdňován pomocí pravidel tvaru

$$\overline{A}|Af \rightarrow \varepsilon|\varepsilon f$$

zkonstruovaných v bodu 6. Podrobně je tato část dokázána v tvrzeních G a H. Z tohoto je zřejmé, že levý vrchol zásobníku musí obsahovat vždy takový symbol s pruhem, který je na pravém vrcholu bez pruhu. Pokud by tomu tak nebylo a zásobník by stále nebyl prázdný, signalizovalo by to nesprávnou simulaci derivace z původní levě rozšířené frontové gramatiky a vstupní řetězec by byl automatem odmítnut. Stejně tak by byl vstupní řetězec odmítnut, pokud by automat nebyl schopen přejít do koncového stavu.

Tvrzení D tedy platí. $\square$

Tvrzení E. Jestliže

$$\begin{array}{ll} & Z_L\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle u,1\rangle\omega \\ \vdash^i & Z_L\overline{B_i}\ldots\overline{B_1}\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mx_1\ldots x_iZ_R\langle p,1\rangle\omega \end{array}$$

v M, pak

$$\begin{array}{ll} & A_1\ldots A_n\#B_1\ldots B_mu \\ \Rightarrow^i & A_1\ldots A_nB_1\ldots B_i\#B_{i+1}\ldots B_mx_1\ldots x_ip \end{array}$$

v G, kde $A_1,\ldots,A_n,B_1,\ldots,B_m \in (V-T)$, $x_1,\ldots,x_i \in (V-T)^*$, $u,p \in (W-F)$, $i < m$.

Základ indukce. Nechť $i = 0$. Potom

$$\begin{array}{ll} & Z_L\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle u,1\rangle\omega \\ \vdash^0 & Z_L\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle u,1\rangle\omega \end{array}$$

v M a

$$A_1\ldots A_n\#B_1\ldots B_mu \Rightarrow^0 A_1\ldots A_n\#B_1\ldots B_mu$$

v G.

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

Indukční krok. Uvažujme v M libovolnou posloupnost přechodů tvaru

$$\begin{array}{ll} & Z_L\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle u,1\rangle\omega \\ \vdash^{l+1} & Z_L\overline{B_{l+1}}\overline{B_l}\ldots\overline{B_1}\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mx_1\ldots x_lx_{l+1}Z_R\langle q,1\rangle\omega \end{array}$$

a rozepišme ji jako

$$\begin{array}{ll} & Z_L\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mZ_R\langle u,1\rangle\omega \\ \vdash^l & Z_L\overline{B_l}\ldots\overline{B_1}\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mx_1\ldots x_lZ_R\langle p,1\rangle\omega \\ \vdash & Z_L\overline{B_{l+1}}\overline{B_l}\ldots\overline{B_1}\overline{A_n}\ldots\overline{A_1}A_1\ldots A_nB_1\ldots B_mx_1\ldots x_lx_{l+1}Z_R\langle q,1\rangle\omega \end{array}$$

kde $q \in (W-F)$, $l+2 \leq m$.

Podle indukční hypotézy

$$\begin{array}{ll} & A_1\ldots A_n\#B_1\ldots B_mu \\ \Rightarrow^l & A_1\ldots A_nB_1\ldots B_l\#B_{l+1}\ldots B_mx_1\ldots x_lp \\ \Rightarrow & A_1\ldots A_nB_1\ldots B_lB_{l+1}\#B_{l+2}\ldots B_mx_1\ldots x_lx_{l+1}q \end{array}$$

v G. V R existuje pouze jeden typ pravidel schopných provést přechod

$$\begin{array}{ll} & Z_L\overline{B_l}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mx_1\dots x_lZ_R\langle p,1\rangle\omega \\ \vdash & Z_L\overline{B_{l+1}}\overline{B_l}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mx_1\dots x_lx_{l+1}Z_R\langle q,1\rangle\omega \end{array}$$

v M. Jsou to pravidla tvaru

$$Z_L|Z_R\langle p,1\rangle \rightarrow Z_L\overline{B_{l+1}}|x_{l+1}Z_R\langle q,1\rangle \in R.$$

Podle bodu 2 konstrukce obsahuje P pravidlo (B_{l+1},p,x_{l+1},q), takže

$$\begin{array}{ll} & A_1\dots A_n\#B_1\dots B_mu \\ \Rightarrow^l & A_1\dots A_nB_1\dots B_l\#B_{l+1}\dots B_mx_1\dots x_lp \\ \Rightarrow & A_1\dots A_nB_1\dots B_lB_{l+1}\#B_{l+2}\dots B_mx_1\dots x_lx_{l+1}q \end{array}$$

v G a Tvrzení E platí. □

Tvrzení F. Jestliže

$$\begin{array}{ll} & Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j \\ \vdash^i & Z_L\overline{B_i}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle p,2\rangle b_{i+1}\dots b_j \end{array}$$

v M, potom

$$A_1\dots A_n\#B_1\dots B_ma_1\dots a_ku \Rightarrow^i A_1\dots A_nB_1\dots B_i\#B_{i+1}\dots B_ma_1\dots a_kb_1\dots b_ip$$

v G, kde $A_1,\dots,A_n,B_1,\dots,B_m \in (V-T)$, $a_1,\dots,a_k,b_1,\dots,b_j \in T^*$ a $p,u \in (W-F)$, $i < m$, $k \geq 0$.

Základ indukce. Nechť $i = 0$. Potom

$$\begin{array}{ll} & Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j \\ \vdash^0 & Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j \end{array}$$

v M a rovněž také

$$A_1\dots A_n\#B_1\dots B_ma_1\dots a_ku \Rightarrow^0 A_1\dots A_n\#B_1\dots B_ma_1\dots a_ku$$

v G.

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

Indukční krok. Uvažujme v M libovolnou posloupnost přechodů tvaru

$$\begin{array}{ll} & Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j \\ \vdash^{l+1} & Z_L\overline{B_{l+1}}\overline{B_l}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle q,2\rangle b_{l+2}\dots b_j \end{array}$$

a vyjádřeme ji jako

$$\begin{array}{ll} & Z_L\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle u,2\rangle b_1\dots b_j \\ \vdash^l & Z_L\overline{B_l}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle p,2\rangle b_{l+1}\dots b_j \\ \vdash & Z_L\overline{B_{l+1}}\overline{B_l}\dots\overline{B_1}\overline{A_n}\dots\overline{A_1}A_1\dots A_nB_1\dots B_mZ_R\langle q,2\rangle b_{l+2}\dots b_j \end{array}$$

kde $l+2 \leq j$, $q \in (W-F)$.

Podle indukční hypotézy

$$\begin{array}{ll} & A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \\ \Rightarrow^l & A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m a_1 \dots a_k b_1 \dots b_l p \\ \Rightarrow & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m a_1 \dots a_k b_1 \dots b_l b_{l+1} q \end{array}$$

v G, kde $l+2 \leq m$.

V tomto případě existuje pouze jediný způsob, kterým může M provést krok

$$\begin{array}{ll} & Z_L \overline{B_l} \dots \overline{B_1 A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle p, 2\rangle b_{l+1} \dots b_j \\ \vdash & Z_L \overline{B_{l+1} B_l} \dots \overline{B_1 A_n} \dots \overline{A_1} A_1 \dots A_n B_1 \dots B_m Z_R \langle q, 2\rangle b_{l+2} \dots b_j. \end{array}$$

Ten je zajištěn pomocí pravidla tvaru

$$Z_L | Z_R \langle p, 2\rangle b_{l+1} \rightarrow Z_L \overline{B_{l+1}} | Z_R \langle q, 2\rangle \in R.$$

Podle bodu 4 konstrukce existuje v P pravidlo (B_{l+1}, p, b_{l+1}, q), kde $B_{l+1} \in (V-T)$, $p, q \in (W-F)$, $b_{l+1} \in T^*$, takže

$$\begin{array}{ll} & A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m a_1 \dots a_k b_1 \dots b_l p \\ \Rightarrow & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m a_1 \dots a_k b_1 \dots b_l b_{l+1} q \end{array}$$

v G a Tvrzení F tedy platí. $\square$

Tvrzení G. Pokud se M nachází v koncovém stavu $f \in F_M$, pak je možné provádět pouze vyprazdňování zásobníku symbol po symbolu.

Důkaz tohoto tvrzení je triviální. Podle bodu 6 konstrukce totiž pouze pravidla tvaru

$$\overline{A} | Af \rightarrow \varepsilon | \varepsilon f$$

odebírají ze zásobníku vždy levý a pravý vrchol, které musí vzájemně korespondovat (symbol vlevo s pruhem musí být shodný se symbolem vpravo, který je ale bez pruhu).

Tvrzení H. Jestliže je automat v konfiguraci tvaru

$$\overline{A_i A_{i-1}} \dots \overline{A_1} A_1 \dots A_{i-1} A_i f \varepsilon,$$

pak je možné v i krocích vyprázdnit zásobník, kde $i \geq 0$.

Základ indukce. Pro $i = 0$ je již zásobník prázdný, tedy automat je v konfiguraci $\varepsilon f \varepsilon$.

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

Indukční krok. Uvažujme libovolnou konfiguraci tvaru

$$\overline{A_{l+1} A_l} \dots \overline{A_1} A_1 \dots A_l A_{l+1} f \varepsilon.$$

Podle indukční hypotézy je zásobník vyprázdněn posloupností přechodů

$$\overline{A_{l+1}A_l}\ldots\overline{A_1}A_1\ldots A_lA_{l+1}f\varepsilon \vdash^l \overline{A_1}A_1f\varepsilon \vdash \varepsilon f\varepsilon.$$

Poslední $l+1$. krok je zajištěn pravidlem tvaru

$$\overline{A_1}|A_1f \rightarrow \varepsilon|\varepsilon f,$$

které bylo zkonstruováno v bodu 6. Po provedení tohoto kroku je již zásobník prázdný. Skutečnost, že musí při vyprazdňování zásobníku v každém okamžiku být levý vrchol tvaru $\overline{A}$ a pravý vrchol tvaru A, kde $A \in Z$, je dáno tvarem pravidel zkonstruovaných v bodu 6. Pokud by si symboly v některém kroku na vrcholech zásobníku neodpovídaly, byl by proces vyprazdňování zablokován, neboť by nebylo aplikovatelné žádné jiné pravidlo. Vstupní řetězec by tedy byl odmítnut.

Tvrzení D, E, F, G a H dohromady dokazují inkluzi $L(M) \subseteq L(G)$. Celkově tedy platí i rovnost $L(G) = L(M)$. Zbývá dokázat poslední tvrzení.

Tvrzení I. Každá věta $w \in L(M)$ je přijímána s jedinou obrátkou zásobníku.

Jak již bylo dokázáno Tvrzením D a jak je ostatně zřejmé z konstrukce, automat používá pravidla z kroku b vždy dříve, než použije pravidla z kroku $b+1$, kde $b = 1, 2, \ldots, 5$. Podle tvarů pravidel zkonstruovaných v krocích 1 až 4 je zřejmé, že je použitím každého takového pravidla zásobník prodloužen (s výjimkou použití pravidla z bodu 3, kdy zůstává jeho délka nezměněna). První zkrácení nastane až použitím některého pravidla zkonstruovaného v bodu 5, kdy je automat převeden do koncového stavu. V tento okamžik tedy M provádí obrátku. Po vstoupení do stavu f je pak možné použít pouze pravidla zkonstruovaná v bodu 6, která zásobník vyprazdňují. Rozhodně jej tedy znovu nemohou prodloužit, tudíž obrátka provedena aplikací některého pravidla z bodu 5 je skutečně jediná.

Tímto je již Věta 4.2 dokázána. ∎

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. Následující věta dokazuje, že třída jazyků definovaná těmito automaty je rovněž shodná s třídou rekurzívně vyčíslitelných jazyků. Přestože je důkaz analogický důkazu předchozímu, uvedeme si zde pro úplnost jeho plné znění.

Věta 4.3 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 $1|1z \to 1|h(S)1\langle q_0, 1\rangle$ 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 $1|1\langle q, 1\rangle \to 1\overline{h}(A)|h(x)1\langle p, 1\rangle$ do R
3) pro každé $q \in W$ přidej $1|1\langle q, 1\rangle \to 1|1\langle q, 2\rangle$ 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 $1|1\langle q, 2\rangle y \to 1\overline{h}(A)|1\langle p, 2\rangle$ 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 $1|1\langle q, 2\rangle y \to \overline{h}(A)|\varepsilon f$ do R
6) přidej do R pravidla $\overline{0}|0f \to \varepsilon|\varepsilon f$ a $\overline{1}|1f \to \varepsilon|\varepsilon f$

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.: Stejně jako v případě neredukované varianty oboustranného zásobníkového automatu si i zde nejprve uvedeme praktický příklad, ve kterém uvedeme podrobnou konstrukci automatu a simulaci přijetí vstupního řetězce. Přestože se budeme již po několikáté opakovat, uvedeme i zde pro úplnost definici výchozí levě rozšířené frontové gramatiky, ze které budeme provádět konstrukci.

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

Množina pravidel této gramatiky obsahuje následující pravidla.

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

Postupnou aplikací dříve uvedených kroků zkonstruujeme oboustranný zásobníkový automat s redukovaným počtem symbolů zásobníkové abecedy

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

kde

- $Q = \{f, z, \langle p, 1\rangle, \langle p, 2\rangle, \langle q, 1\rangle, \langle q, 2\rangle\}$
- $T = \{a, b\}$
- $Z = \{0, \overline{0}, 1, \overline{1}\}$
- $F_M = \{f\}$
- *symboly z množiny* $(V - T)$ *zakódujeme takto:*

	A	B
h	000	010
$\overline{h}$	$\overline{000}$	$\overline{010}$

- *konstrukce množiny pravidel* R *je uvedena v následující tabulce*

Označení		***Pravidlo***		***Odp. pravidlo z*** Q
1a)	$1\|1z$	$\rightarrow$	$1\|0001\langle p, 1\rangle$	—
2a)	$1\|1\langle p, 1\rangle$	$\rightarrow$	$1\overline{000}\|0000001\langle p, 1\rangle$	(A, p, AA, p)
2b)	$1\|1\langle p, 1\rangle$	$\rightarrow$	$1\overline{000}\|0100101\langle p, 1\rangle$	(A, p, BB, p)
3a)	$1\|1\langle p, 1\rangle$	$\rightarrow$	$1\|1\langle p, 2\rangle$	—
3b)	$1\|1\langle q, 1\rangle$	$\rightarrow$	$1\|1\langle q, 2\rangle$	—
3c)	$1\|1\langle f, 1\rangle$	$\rightarrow$	$1\|1\langle f, 2\rangle$	—
4a)	$1\|1\langle p, 2\rangle a$	$\rightarrow$	$1\overline{000}\|1\langle q, 2\rangle$	(A, p, a, q)
4b)	$1\|1\langle q, 2\rangle a$	$\rightarrow$	$1\overline{000}\|1\langle q, 2\rangle$	(A, q, a, q)
4c)	$1\|1\langle q, 2\rangle b$	$\rightarrow$	$1\overline{010}\|1\langle q, 2\rangle$	(B, q, b, q)
5a)	$1\|1\langle q, 2\rangle b$	$\rightarrow$	$\overline{010}\|\varepsilon f$	(B, q, b, f)
6a)	$\overline{0}\|0f$	$\rightarrow$	$\varepsilon\|\varepsilon f$	—
6b)	$\overline{1}\|1f$	$\rightarrow$	$\varepsilon\|\varepsilon f$	—

*Stejně jako v příkladu pro neredukovanou verzi automatu je ve sloupci **Označení** uveden identifikátor každého pravidla, ve kterém číslo odpovídá číslu kroku v konstrukci a písmeno rozlišuje dané pravidlo v rámci kroku konstrukce. Sloupec **Pravidlo** pak obsahuje zkonstruovaná pravidla automatu M a sloupec **Odp. pravidlo z** Q obsahuje korespondující pravidlo z gramatiky Q, ze kterého bylo dané pravidlo M odvozeno.*

I zde připomeňme, že věta $abb \in L(Q)$, jejíž přijetí budeme automatem simulovat, je ve výchozí 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 konečně můžeme uvést posloupnost kroků, kterými tuto větu přijme námi zkonstruovaný oboustranný zásobníkový automat s redukovaným počtem symbolů zásobníkové abecedy M.

$$
\begin{array}{lll}
 & 11zabb & \\
\vdash & 10001\langle p,1\rangle abb & 1a \\
\vdash & 1\overline{000}0000000001\langle p,1\rangle abb & 2a \\
\vdash & 1\overline{000000}0000000000100101\langle p,1\rangle abb & 2b \\
\vdash & 1\overline{000000}0000000000100101\langle p,2\rangle abb & 3a \\
\vdash & 1\overline{000000000}0000000000100101\langle q,2\rangle bb & 4a \\
\vdash & 1\overline{010000000000}0000000000100101\langle q,2\rangle b & 4c \\
\vdash & \overline{010010000000000}000000000010010f\varepsilon & 5a \\
\vdash & \overline{10010000000000}00000000001001f\varepsilon & 6a \\
\vdash & \overline{0010000000000}0000000000100f\varepsilon & 6b \\
\vdash & \overline{010000000000}000000000010f\varepsilon & 6a \\
\vdash & \overline{10000000000}00000000001f\varepsilon & 6a \\
\vdash & \overline{0000000000}0000000000f\varepsilon & 6b \\
\vdash & \overline{000000000}000000000f\varepsilon & 6a \\
\vdash & \overline{00000000}00000000f\varepsilon & 6a \\
\vdash & \overline{0000000}0000000f\varepsilon & 6a \\
\vdash & \overline{000000}000000f\varepsilon & 6a \\
\vdash & \overline{00000}00000f\varepsilon & 6a \\
\vdash & \overline{0000}0000f\varepsilon & 6a \\
\vdash & \overline{000}000f\varepsilon & 6a \\
\vdash & \overline{00}00f\varepsilon & 6a \\
\vdash & \overline{0}0f\varepsilon & 6a \\
\vdash & \varepsilon f\varepsilon & 6a
\end{array}
$$

I v tomto případě skončil M v konfiguraci $\varepsilon f\varepsilon$ a vstupní řetězec byl úspěšně přijat. Z tohoto přístupu jsou však zřejmé dvě nevýhody. Tou první je nutnost alokovat několikanásobně větší prostor na zásobníku než by tomu bylo u neredukované verze automatu. Druhou nevýhodou je nižší přehlednost. Výhodou je naopak nízký počet symbolů zásobníkové abecedy.

Rigorózní důkaz

Nyní je třeba formálně dokázat rovnost $L(G) = L(M)$, tedy $L(G) \subseteq L(M)$ a $L(M) \subseteq L(G)$. Demonstrací Tvrzení A, B a C nejprve dokážeme $L(G) \subseteq L(M)$.

Tvrzení A. Jestliže

$$A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^i A_1 \dots A_n B_1 \dots B_i \# B_{i+1} \dots B_m x_1 \dots x_i p$$

v G, pak

$$\begin{array}{ll} & 1\overline{h}(A_n) \dots \overline{h}(A_1) h(A_1) \dots h(A_n) h(B_1) \dots h(B_m) 1 \langle u, 1 \rangle \omega \\ \vdash^i & 1\overline{h}(B_i) \dots \overline{h}(B_1) \overline{h}(A_n) \dots \overline{h}(A_1) h(A_1) \dots h(A_n) h(B_1) \dots h(B_m) h(x_1) \dots h(x_i) 1 \langle p, 1 \rangle \omega \end{array}$$

v M, kde $A_1, \dots, A_n, B_1, \dots, B_m \in (V - T)$, $x_1, \dots, x_i \in (V - T)^*$, $u, p \in (W - F)$, $n \geq 0$, $\omega \in T^*$, $0 \leq i < m$.

Základ indukce. Nechť $i = 0$. Potom

$$A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^0 A_1 \dots A_n \# B_1 \dots B_m u$$

v G a

$$\begin{array}{ll} & 1\overline{h}(A_n) \dots \overline{h}(A_1) h(A_1) \dots h(A_n) h(B_1) \dots h(B_m) 1 \langle u, 1 \rangle \omega \\ \vdash^0 & 1\overline{h}(A_n) \dots \overline{h}(A_1) h(A_1) \dots h(A_n) h(B_1) \dots h(B_m) 1 \langle u, 1 \rangle \omega \end{array}$$

v M.

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

Indukční krok. Uvažujme libovolnou derivaci tvaru

$$A_1 \dots A_n \# B_1 \dots B_m u \Rightarrow^{l+1} A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m x_1 \dots x_l x_{l+1} q$$

a vyjádřeme ji jako

$$\begin{array}{ll} & A_1 \dots A_n \# B_1 \dots B_m u \\ \Rightarrow^l & A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m x_1 \dots x_l p \\ \Rightarrow & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m x_1 \dots x_l x_{l+1} q \end{array}$$

v G, kde $(l + 2) \leq m$, $q \in (W - F)$.

Podle indukční hypotézy

$$\begin{array}{ll} & 1\overline{h}(A_n) \dots \overline{h}(A_1) h(A_1) \dots h(A_n) h(B_1) \dots h(B_m) 1 \langle u, 1 \rangle \omega \\ \vdash^l & 1\overline{h}(B_l) \dots \overline{h}(B_1) \overline{h}(A_n) \dots \overline{h}(A_1) h(A_1) \dots h(A_n) h(B_1) \dots h(B_m) h(x_1) \dots h(x_l) 1 \langle p, 1 \rangle \omega \\ \vdash & 1\overline{h}(B_{l+1}) \overline{h}(B_l) \dots \overline{h}(B_1) \overline{h}(A_n) \dots \overline{h}(A_1) h(A_1) \dots h(A_n) h(B_1) \dots h(B_m) h(x_1) \dots \\ & \dots h(x_l) h(x_{l+1}) 1 \langle q, 1 \rangle \omega \end{array}$$

v M. Pomocí pravidla tvaru $(B_{l+1}, p, x_{l+1}, q) \in P$, kde $B_{l+1} \in (V - T)$, $p, q \in (W - F)$ a $x_{l+1} \in (V - T)^*$, je provedena derivace

$$\begin{array}{rl} & A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m x_1 \dots x_l p \\ \Rightarrow & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m x_1 \dots x_l x_{l+1} q \end{array}$$

v G. Podle bodu 2 konstrukce existuje v R pravidlo

$$1|1\langle p, 1\rangle \rightarrow 1\overline{h}(B_{l+1})|h(x_{l+1})1\langle q, 1\rangle,$$

takže

$$\begin{array}{rl} & 1\overline{h}(B_l) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)h(x_1) \dots h(x_l)1\langle p, 1\rangle\omega \\ \vdash & 1\overline{h}(B_{l+1})\overline{h}(B_l) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)h(x_1) \dots \\ & \dots h(x_l)h(x_{l+1})1\langle q, 1\rangle\omega \end{array}$$

v M a Tvrzení A platí. □

Tvrzení B. Jestliže

$$\begin{array}{rl} & A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \\ \Rightarrow^i & A_1 \dots A_n B_1 \dots B_i \# B_{i+1} \dots B_m a_1 \dots a_k b_1 \dots b_i p \end{array}$$

v G, pak

$$\begin{array}{rl} & 1\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle u, 2\rangle b_1 \dots b_j \\ \vdash^i & 1\overline{h}(B_i) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots \\ & \dots h(B_m)1\langle p, 2\rangle b_{i+1} \dots b_j \end{array}$$

v M, kde $A_1, \dots, A_n, B_1, \dots, B_m \in (V - T)$, $a_1, \dots, a_k$, $b_1, \dots, b_j \in T^*$, $u, p \in (W - F)$, $k \geq 0$, $0 \leq i < j \leq m$.

Základ indukce. Nechť $i = 0$. Potom

$$A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \Rightarrow^0 A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u$$

v G. Rozhodně také

$$\begin{array}{rl} & 1\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle u, 2\rangle b_1 \dots b_j \\ \vdash^0 & 1\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle u, 2\rangle b_1 \dots b_j \end{array}$$

v M.

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

Indukční krok. Uvažujme libovolnou derivaci tvaru

$$\begin{array}{rl} & A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \\ \Rightarrow^{l+1} & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m a_1 \dots a_k b_1 \dots b_l b_{l+1} q \end{array}$$

a vyjádřeme ji přesněji jako

$$\begin{array}{rl} & A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \\ \Rightarrow^l & A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m a_1 \dots a_k b_1 \dots b_l p \\ \Rightarrow & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m a_1 \dots a_k b_1 \dots b_l b_{l+1} q \end{array}$$

v G, kde $k \geq 0$, $0 \leq (l+2) \leq m$, $q \in (W - F)$.

Podle indukční hypotézy

$$\begin{array}{ll} & 1\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle u,2\rangle b_1\ldots b_j \\ \vdash^l & 1\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle p,2\rangle b_{l+1}\ldots b_j \\ \vdash & 1\overline{h}(B_{l+1})\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle q,2\rangle b_{l+2}\ldots b_j \end{array}$$

v M.

Derivace

$$\begin{array}{ll} & A_1\ldots A_nB_1\ldots B_l\#B_{l+1}\ldots B_ma_1\ldots a_kb_1\ldots b_lp \\ \Rightarrow & A_1\ldots A_nB_1\ldots B_lB_{l+1}\#B_{l+2}\ldots B_ma_1\ldots a_kb_1\ldots b_lb_{l+1}q \end{array}$$

v G je provedena pomocí pravidla tvaru $(B_{l+1}, p, b_{l+1}, q) \in P$, kde $B_{l+1} \in (V - T)$, $p, q \in (W - F)$, $b_{l+1} \in T^*$. Podle bodu 4 konstrukce existuje v R pravidlo

$$1|1\langle p,2\rangle b_{l+1} \to 1\overline{h}(B_{l+1})|1\langle q,2\rangle,$$

takže

$$\begin{array}{ll} & 1\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle p,2\rangle b_{l+1}\ldots b_j \\ \vdash & 1\overline{h}(B_{l+1})\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle q,2\rangle b_{l+2}\ldots b_j \end{array}$$

v M a Tvrzení B tedy platí. □

Tvrzení C. Jestliže

$$A_1\ldots A_{n-1}\#A_nyq \Rightarrow A_1\ldots A_{n-1}A_n\#yzt$$

v G, kde $A_1, \ldots, A_n \in (V - T)$, $y, z \in T^*$, $q \in (W - F)$, $t \in F$, pak

$$1\overline{h}(A_{n-1})\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)1\langle q,2\rangle z \vdash \overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)f$$

v M, kde $f \in F_M$.

Gramatika G provádí uvedenou derivaci pomocí pravidla tvaru $(A_n, q, z, t) \in P$, kde $A_n \in (V - T)$, $z \in T^*$, $q \in (W - F)$, $t \in F$. Podle bodu 5 konstrukce existuje v R pravidlo

$$1|1\langle q,2\rangle z \to \overline{h}(A_n)|\varepsilon f,$$

takže

$$1\overline{h}(A_{n-1})\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)1\langle q,2\rangle z \vdash \overline{h}(A_n)\overline{h}(A_{n-1})\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)f\varepsilon$$

v M a Tvrzení C tedy platí. □

Tvrzení A, B a C dokazují, že $L(G) \subseteq L(M)$. V další části dokážeme platnost obrácené inkluze, tedy $L(M) \subseteq L(G)$.

Tvrzení D. Automat M přijímá každou větu $w \in L(M)$ následujícím způsobem.

$11zw_1w_2\dots w_r \vdash$
$1h(S)1\langle q_0,1\rangle w_1w_2\dots w_r \vdash$
$1\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots h(X_{n_1}^1)1\langle q_1,1\rangle w_1w_2\dots w_r \vdash$
$1\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)1\langle q_2,1\rangle w_1w_2\dots w_r \vdash$
$1\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\dots$
$h(X_{n_3}^3)1\langle q_3,1\rangle w_1w_2\dots w_r \vdash$

$\dots$

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

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

$1\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots$
$h(X_{n_2}^2)h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m}^m)1\langle q_{m+1},2\rangle w_2\dots w_r \vdash$

$1\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots h(X_{n_1}^1)h(X_1^2)h(X_2^2)$
$\dots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m}^m)1\langle q_{m+2},2\rangle w_3\dots w_r \vdash$

$\dots$

$1\overline{h}(X_{n_m-1}^m)\dots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots$
$\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m}^m)1\langle q_{m+r-1},2\rangle w_r \vdash$

$\overline{h}(X_{n_m}^m)\overline{h}(X_{n_m-1}^m)\dots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots$
$\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m}^m)f\varepsilon \vdash^{n+2}$

$\overline{h}(X_{n_m-1}^m)\dots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots$
$\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m-1}^m)f\varepsilon \vdash^{n+2}$

$\overline{h}(X_{n_m-2}^m)\dots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots$
$\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m-2}^m)f\varepsilon \vdash^{n+2}$

$\dots$

$\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)f \vdash^{n+2}$
$\overline{h}(S)h(S)f \vdash^{n+2}$
$\varepsilon f\varepsilon,$

kde $w = w_1w_2\dots w_r$, $r \geq 1$, $w_1, \dots, w_r \in T^*$, $q_0, q_1, \dots, q_{m+r-1} \in (W-F)$, $X_1^1, \dots, X_{n_1}^1, X_1^2, \dots, X_{n_2}^2, \dots, X_1^m, \dots, X_{n_m}^m \in (V-T)$, $n_1, n_2, \dots, n_m \geq 0$, $0 \leq k \leq m$.

Prozkoumáme všechny kroky konstrukce množiny pravidel R. Připomeňme, že automat používá v každé úspěšné posloupnosti přechodů pravidla zkonstruovaná v kroku b před tím, než použije pravidla zkonstruovaná v kroku $b+1$, kde $b = 1, \dots, 5$.

V prvním kroku aplikuje M pravidlo

$$1|1z \to 1|h(S)1\langle q_0, 1\rangle$$

zkonstruované v bodu 1, kde Sq_0 je výchozí axiom gramatiky G. Toto je jediný způsob, kterým M může provést přechod

$$11zw_1w_2\dots w_r \vdash 1h(S)1\langle q_0, 1\rangle w_1w_2\dots w_r$$

a přepnout se tak do módu generování nonterminálů. Všimněme si, že toto pravidlo je v každé úspěšné posloupnosti přechodů použito právě jednou.

V posloupnosti přechodů

$1h(S)1\langle q_0, 1\rangle w_1w_2\dots w_r \vdash^*$
$1\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)$
$h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m}^m)1\langle q_m, 1\rangle w_1w_2\dots w_r$

používá M pravidla tvaru

$$1|1\langle q, 1\rangle \to 1\overline{h}(A)|h(x)1\langle p, 1\rangle$$

zkonstruovaná v bodu 2, kde $A \in (V-T)$, $x \in (V-T)^*$, $p, q \in (W-F)$. Tato posloupnost je charakterizována stavy automatu tvaru $\langle q, 1\rangle$, kde $q \in (W-F)$. Detailní důkaz této části popisuje Tvrzení E.

Krokem

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

$1\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)$
$h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m}^m)1\langle q_m, 2\rangle w_1w_2\dots w_r$

je M přepnut do módu čtení terminálů. To je zajištěno použitím některého z pravidel tvaru

$$1|1\langle q, 1\rangle \to 1|1\langle q, 2\rangle$$

zkonstruovaných v bodu 3. Pravidlo tohoto typu je použito v průběhu jedné úspěšné výpočetní posloupnosti právě jednou. Protože toto pravidlo mění stav automatu tvaru $\langle q, 1\rangle$ na stav tvaru $\langle q, 2\rangle$, pro aktuální $q \in (W-F)$, není poté již žádná další možnost použití pravidel zkonstruovaných v bodech 1 až 3.

V části

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

$1\overline{h}(X_{n_m-1}^m)\dots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\dots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\dots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\dots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\dots h(X_{n_3}^3)\dots$
$\dots h(X_1^m)h(X_2^m)\dots h(X_{n_m}^m)1\langle q_{m+r-1}, 2\rangle w_r$

používá M pravidla zkonstruovaná v bodu 4 a postupně čte vstupní řetězec tvořený terminálními symboly. Detailní důkaz této části je uveden v Tvrzení F.

V přechodu

$1\overline{h}(X_{n_m-1}^m)\ldots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\ldots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\ldots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\ldots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\ldots h(X_{n_3}^3)\ldots$
$\ldots h(X_1^m)h(X_2^m)\ldots h(X_{n_m}^m)1\langle q_{m+r-1},2\rangle w_r \vdash$

$\overline{h}(X_{n_m}^m)\overline{h}(X_{n_m-1}^m)\ldots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\ldots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\ldots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\ldots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\ldots h(X_{n_3}^3)\ldots$
$\ldots h(X_1^m)h(X_2^m)\ldots h(X_{n_m}^m)f\varepsilon$

je M přiveden do koncového stavu. K tomuto účelu je použito pravidlo tvaru

$$1|1\langle q,2\rangle y \rightarrow \overline{h}(A)|\varepsilon f$$

zkonstruované v bodu 5, kde $q \in (W-T)$, $y \in T^*$, $A \in (V-T)$ a $f \in F_M$.

Nyní zbývá splnit poslední podmínku pro přijetí řetězce, a to vyprázdnit zásobník. Postupné vyprazdňování zásobníku (po kódových řetězcích jednotlivých symbolů z $(V-T)$) znázorňuje posloupnost přechodů

$\overline{h}(X_{n_m}^m)\overline{h}(X_{n_m-1}^m)\ldots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\ldots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\ldots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\ldots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\ldots h(X_{n_3}^3)\ldots$
$\ldots h(X_1^m)h(X_2^m)\ldots h(X_{n_m}^m)f\varepsilon \vdash^{n+2}$

$\overline{h}(X_{n_m-1}^m)\ldots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\ldots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\ldots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\ldots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\ldots h(X_{n_3}^3)\ldots$
$\ldots h(X_1^m)h(X_2^m)\ldots h(X_{n_m-1}^m)f\varepsilon \vdash^{n+2}$

$\overline{h}(X_{n_m-2}^m)\ldots\overline{h}(X_{j+2}^k)\overline{h}(X_{j+1}^k)\overline{h}(X_j^k)\ldots\overline{h}(X_2^1)\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)h(X_2^1)\ldots$
$h(X_{n_1}^1)h(X_1^2)h(X_2^2)\ldots h(X_{n_2}^2)h(X_1^3)h(X_2^3)\ldots h(X_{n_3}^3)\ldots$
$\ldots h(X_1^m)h(X_2^m)\ldots h(X_{n_m-2}^m)f\varepsilon \vdash^{n+2}$

$\ldots$

$\overline{h}(X_1^1)\overline{h}(S)h(S)h(X_1^1)f\varepsilon \vdash^{n+2}$
$\overline{h}(S)h(S)f\varepsilon \vdash^{n+2}$
$\varepsilon f\varepsilon.$

Jednotlivé symboly jsou odstraňovány z levého i pravého vrcholu zásobníku pomocí pravidel

$$\overline{0}|0f \rightarrow \varepsilon|\varepsilon f \quad \text{a} \quad \overline{1}|1f \rightarrow \varepsilon|\varepsilon f$$

zkonstruovaných v bodu 6. Opět jako v případě neredukované verze oboustranného zásobníkového automatu je nutné, aby levý symbol s pruhem a pravý symbol vzájemně korespondovaly. Jedině tak je možné úspěšně zásobník vyprázdnit. Podrobně popisují vyprazdňování zásobníku Tvrzení G a H. Tímto je Tvrzení D dokázáno.

Tvrzení E. Jestliže

$$\begin{array}{ll} & 1\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle u,1\rangle\omega \\ \vdash^i & 1\overline{h}(B_i)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)h(x_1)\ldots h(x_i)1\langle p,1\rangle\omega \end{array}$$

v M, potom

$$A_1\ldots A_n\#B_1\ldots B_m u \Rightarrow^i A_1\ldots A_n B_1\ldots B_i\#B_{i+1}\ldots B_m x_1\ldots x_i p$$

v G, kde $A_1,\ldots,A_n,B_1,\ldots,B_m \in (V-T)$, $x_1,\ldots,x_i \in (V-T)^*$, $u,p \in (W-F)$, $0 \leq i < m$.

Základ indukce. Nechť $i = 0$. Potom

$$\begin{array}{ll} & 1\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle u,1\rangle\omega \\ \vdash^0 & 1\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle u,1\rangle\omega \end{array}$$

v M a

$$A_1\ldots A_n\#B_1\ldots B_m u \Rightarrow^0 A_1\ldots A_n\#B_1\ldots B_m u$$

v G.

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

Indukční krok. Uvažujme libovolnou posloupnost přechodů tvaru

$$\begin{array}{ll} & 1\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle u,1\rangle\omega \\ \vdash^{l+1} & 1\overline{h}(B_{l+1})\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)h(x_1)\ldots \\ & \ldots h(x_l)h(x_{l+1})1\langle q,1\rangle\omega \end{array}$$

v M a vyjádřeme ji přesněji jako

$$\begin{array}{ll} & 1\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)1\langle u,1\rangle\omega \\ \vdash^l & 1\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)h(x_1)\ldots h(x_l)1\langle p,1\rangle\omega \\ \vdash & 1\overline{h}(B_{l+1})\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)h(x_1)\ldots \\ & \ldots h(x_l)h(x_{l+1})1\langle q,1\rangle\omega, \end{array}$$

kde $q \in (W-F)$, $0 \leq l \leq m$.

Podle indukční hypotézy

$$\begin{array}{ll} & A_1\ldots A_n\#B_1\ldots B_m u \\ \Rightarrow^l & A_1\ldots A_n B_1\ldots B_l\#B_{l+1}\ldots B_m x_1\ldots x_l p \\ \Rightarrow & A_1\ldots A_n B_1\ldots B_l B_{l+1}\#B_{l+2}\ldots B_m x_1\ldots x_l x_{l+1} q \end{array}$$

v G. Přechod

$1\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)h(x_1)\ldots h(x_l)1\langle p,1\rangle\omega \vdash$
$1\overline{h}(B_{l+1})\overline{h}(B_l)\ldots\overline{h}(B_1)\overline{h}(A_n)\ldots\overline{h}(A_1)h(A_1)\ldots h(A_n)h(B_1)\ldots h(B_m)h(x_1)\ldots$
$\ldots h(x_l)h(x_{l+1})1\langle q,1\rangle\omega$

v M je proveden pomocí pravidla tvaru

$$1|1\langle p,1\rangle \rightarrow 1\overline{h}(B_{l+1})|h(x_{l+1})1\langle q,1\rangle \in R.$$

Podle bodu 2 konstrukce ale obsahuje P pravidlo (B_{l+1}, p, x_{l+1}, q), takže

$$\begin{array}{ll} & A_1 \dots A_n \# B_1 \dots B_m u \\ \Rightarrow^l & A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m x_1 \dots x_l p \\ \Rightarrow & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m x_1 \dots x_l x_{l+1} q \end{array}$$

v G a Tvrzení E platí. □

Tvrzení F. Jestliže

$$\begin{array}{ll} & 1\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle u,2\rangle b_1 \dots b_j \\ \vdash^i & 1\overline{h}(B_i) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_i)h(B_{i+1}) \dots \\ & \dots h(B_m)1\langle p,2\rangle b_{i+1} \dots b_j \end{array}$$

v M, pak

$$\begin{array}{ll} & A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \\ \Rightarrow^i & A_1 \dots A_n B_1 \dots B_i \# B_{i+1} \dots B_m a_1 \dots a_k b_1 \dots b_i p \end{array}$$

v G, kde $A_1, \dots, A_n, B_1, \dots, B_m \in (V-T)$, $a_1, \dots, a_k$, $b_1, \dots, b_j \in T^*$ a $p, u \in (W-F)$, $0 \leq i < m$, $k \geq 0$.

Základ indukce. Nechť $i = 0$. Potom

$$\begin{array}{ll} & 1\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle u,2\rangle b_1 \dots b_j \\ \vdash^0 & 1\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle u,2\rangle b_1 \dots b_j \end{array}$$

v M a určitě také

$$A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \Rightarrow^0 A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u$$

v G.

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

Indukční krok. Uvažujme libovolnou posloupnost přechodů tvaru

$$\begin{array}{ll} & 1\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle u,2\rangle b_1 \dots b_j \\ \vdash^{l+1} & 1\overline{h}(B_{l+1})\overline{h}(B_l) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle q,2\rangle b_{l+2} \dots b_j \end{array}$$

v M a vyjádřeme ji jako

$$\begin{array}{ll} & 1\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle u,2\rangle b_1 \dots b_j \\ \vdash^l & 1\overline{h}(B_l) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle p,2\rangle b_{l+1} \dots b_j \\ \vdash & 1\overline{h}(B_{l+1})\overline{h}(B_l) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle q,2\rangle b_{l+2} \dots b_j, \end{array}$$

kde $0 \leq (l+1) \leq j \leq m$, $q \in (W-F)$.

Podle indukční hypotézy

$$
\begin{aligned}
& A_1 \dots A_n \# B_1 \dots B_m a_1 \dots a_k u \Rightarrow^l \\
\Rightarrow^l \quad & A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m a_1 \dots a_k b_1 \dots b_l p \\
\Rightarrow \quad & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m a_1 \dots a_k b_1 \dots b_l b_{l+1} q
\end{aligned}
$$

v G, kde $0 \leq (l+2) \leq m$, $k \geq 0$.

V tomto případě provede M krok

$$
\begin{aligned}
& 1\overline{h}(B_l) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle p, 2\rangle b_{l+1} \dots b_j \\
\vdash \quad & 1\overline{h}(B_{l+1})\overline{h}(B_l) \dots \overline{h}(B_1)\overline{h}(A_n) \dots \overline{h}(A_1)h(A_1) \dots h(A_n)h(B_1) \dots h(B_m)1\langle q, 2\rangle b_{l+2} \dots b_j
\end{aligned}
$$

použitím pravidla tvaru

$$
1|1\langle p, 2\rangle b_{l+1} \rightarrow 1\overline{h}(B_{l+1})|1\langle q, 2\rangle \in R.
$$

Podle bodu 4 konstrukce existuje v P pravidlo (B_{l+1}, p, b_{l+1}, q), kde $B_{l+1} \in (V - T)$, $p, q \in (W - F)$, $b_{l+1} \in T^*$, takže

$$
\begin{aligned}
& A_1 \dots A_n B_1 \dots B_l \# B_{l+1} \dots B_m a_1 \dots a_k b_1 \dots b_l p \\
\Rightarrow \quad & A_1 \dots A_n B_1 \dots B_l B_{l+1} \# B_{l+2} \dots B_m a_1 \dots a_k b_1 \dots b_l b_{l+1} q
\end{aligned}
$$

v G a Tvrzení F platí. $\square$

Tvrzení G. Pokud se M nachází v koncovém stavu f, pak je možné provádět pouze vyprazdňování zásobníku symbol po symbolu.

Důkaz je triviální. Podle bodu 6 konstrukce obsahuje M pravidla

$$
\overline{0}|0f \rightarrow \varepsilon|\varepsilon f \quad \text{a} \quad \overline{1}|1f \rightarrow \varepsilon|\varepsilon f,
$$

která postupně odebírají vždy dva korespondující symboly z levého a pravého vrcholu zásobníku (je-li levý vrchol $\overline{0}$ resp. $\overline{1}$, musí být pravý vrchol nutně 0 resp. 1).

Tvrzení H. Nechť M kóduje nonterminální symboly gramatiky G řetězci o pevné délce k symbolů. Bez ztráty obecnosti můžeme předpokládat, že $k = n + 2$, kde

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

Nechť se dále automat nachází v konfiguraci

$$
\overline{h}(A_i)\overline{h}(A_{i-1}) \dots \overline{h}(A_1)h(A_1) \dots h(A_{i-1})h(A_i)f\varepsilon.
$$

Pak je možné vyprázdnit zásobník v ki krocích, pro $i \geq 0$.

Základ indukce. Pro $i = 0$ je již zásobník prázdný, tedy automat se nachází v konfiguraci $\varepsilon f \varepsilon$.

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

Indukční krok. Uvažujme libovolnou konfiguraci tvaru

$$
\overline{h}(A_{l+1})\overline{h}(A_l)\overline{h}(A_{l-1}) \dots \overline{h}(A_1)h(A_1) \dots h(A_{l-1})h(A_l)h(A_{l+1})f\varepsilon.
$$

Podle indukční hypotézy je zásobník vyprázdněn posloupností přechodů

$$\overline{h}(A_{l+1})\overline{h}(A_l)\overline{h}(A_{l-1})\ldots\overline{h}(A_1)h(A_1)\ldots h(A_{l-1})h(A_l)h(A_{l+1})f \vdash^{kl} \overline{h}(A_1)h(A_1)f \vdash^{k} \varepsilon f\varepsilon.$$

Poslední část celé posloupnosti má délku k přechodů a je možné ji přesněji rozepsat jako

$$\begin{array}{ll} & \overline{c_k c_{k-1}}\ldots\overline{c_1}c_1\ldots c_{k-1}c_k f\varepsilon \\ \vdash & \overline{c_{k-1}}\ldots\overline{c_1}c_1\ldots c_{k-1}f\varepsilon \\ \vdash^{k-1} & \varepsilon f\varepsilon, \end{array}$$

kde $\overline{c_1},\ldots,\overline{c_k} \in \{\overline{0},\overline{1}\}$ a $c_1,\ldots,c_k \in \{0,1\}$. Podle bodu 6 konstrukce musela být použita pravidla

$$\overline{0}|0f \to \varepsilon|\varepsilon f \quad \text{a} \quad \overline{1}|1f \to \varepsilon|\varepsilon f.$$

Nyní je již zásobník prázdný. Zopakujme, že symboly na vrcholech zásobníku musí vzájemně korespondovat. Pokud by tomu tak nebylo, zásobník by nebylo možné vyprázdnit a vstupní řetězec by nebyl přijat.

Tvrzení D, E, F, G a H dokazují, že $L(M) \subseteq L(G)$. Celkově tedy skutečně platí rovnost $L(G) = L(M)$. Poslední tvrzení se týká vlastnosti, kterou je možno charakterizovat každý úspěšný výpočetní proces.

Tvrzení I. Každá věta $w \in L(M)$ je přijímána s jedinou obrátkou zásobníku.

Důkaz tohoto tvrzení je podobný důkazu stejného tvrzení pro neredukovanou verzi oboustranného zásobníkového automatu. Pro úplnost jej ale uveďme. Jak již bylo dokázáno Tvrzením D a jak je ostatně zřejmé z konstrukce, automat používá pravidla z kroku b vždy dříve, než použije pravidla z kroku $b+1$, kde $b = 1, 2, \ldots, 5$. Podle tvarů pravidel zkonstruovaných v krocích 1 až 5 a s ohledem na délku kódových slov jednotlivých nonterminálních symbolů je zřejmé, že je použitím každého takového pravidla zásobník prodloužen (s výjimkou použití pravidla z bodu 3, kdy zůstává jeho délka nezměněna). První zkrácení nastane až prvním použitím některého pravidla zkonstruovaného v bodu 6, kdy začíná být zásobník vyprazdňován. V tento okamžik tedy M provádí obrátku. S ohledem na tvar pravidel v bodu 6 již není možné opětovné prodloužení řetězce na zásobníku a tudíž zmiňovaná obrátka je skutečně jediná.

Nyní je již důkaz Věty 4.3 hotov. ■

4.3 Shrnutí

Výše uvedená konstrukce a důkaz potvrzují, že oboustranné zásobníkové automaty mají stejnou vyjadřovací sílu jako frontové gramatiky a jsou tudíž schopné definovat celou třídu rekurzívně vyčíslitelných jazyků. Srovnáme-li tyto automaty s běžnými zásobníkovými automaty, získali jsme jednoduchou modifikací mnohem silnější formální model. Přitom jsme pouze rozšířili běžný zásobník na oboustrannou variantu a tím umožnili vkládat a odebírat potřebné symboly a řetězce z obou stran. Toto je jediné rozšíření struktury běžného zásobníkového automatu.

Přestože jsou tyto automaty velmi podobné již dlouho známým dvouzásobníkovým automatům, mají i tak několik předností. V první řadě je to celistvost takto definovaného zásobníku. Ta umožňuje definovat zásobník nad volnou grupou a jeho vyprazdňování provádět implicitními redukcemi s využitím inverzních symbolů. Tím odpadne nutnost použití pravidel z bodu 6. Zavedení volné grupy do oboustranných zásobníkových automatů je podrobně studováno v [8]. Další výhodou je možnost transformace oboustranného zásobníku na frontu, kdy není třeba výrazně měnit definici, ale pouze přípustné tvary pravidel.

Na první pohled se může zdát, že jsou tyto automaty velmi jednoduchou modifikací souběžně jednoobrátkových dvouzásobníkových automatů prezentovaných v [22]. I když jsme z tého publikace vycházeli a uvedené automaty byly pro náš výzkum hlavní inspirací, přináší tento důkaz výrazné zjednodušení konstrukce, kdy přímo využívá nonterminální symboly (resp. jejich binární kódy) z frontové gramatiky a s nimi pracuje na zásobníku. V [22] je tato problematika řešena pomocí substitucí, což představuje složitější přístup jak ke konstrukci pravidel automatu, tak i k celému důkazu a činí jej tímto méně přehledným.

Oboustranné jednoobrátkové zásobníkové automaty (a jejich redukované verze) představují další přírůstky do rodiny modelů pro popis formálních jazyků.

Kapitola 5

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.

5.1 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

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

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 $AB \rightarrow CD \in P$, 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 5.1 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

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

kde $n, m \geq 1$, 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 $\langle X, u, \varepsilon, \varepsilon\rangle x \to \langle X, u, x, \varepsilon\rangle$

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

  III pro každé $AB \to CD \in P$ přidej:
  - *(a)* $\langle X, u, \alpha C, v\rangle \to \langle X, u, \alpha A, vAB \to CD\rangle$ do R_{CS1}
  - *(b)* $\langle X, AB \to CDu, D\beta, v\rangle \to \langle X, u, B\beta, v\rangle$ do R_{CS2}
  - *(c)* $\langle X, u, \alpha CD\beta, v\rangle \to \langle X, u, \alpha AB\beta, v\rangle$ do R_{IN}

  IV pro každé $A \to BC \in P$ přidej:
  - *(a)* $\langle X, u, \alpha BC\beta, v\rangle \to \langle X, u, \alpha A\beta, v\rangle$ do R_{IN}
  - *(b)* $\langle A, \varepsilon, B, v\rangle \to \langle C, v, \varepsilon, \varepsilon\rangle$ do R_{HEAD}

  V pro každé $A \in N$, přidej $\langle A, \varepsilon, A, \varepsilon\rangle \to f$ 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

   $$AB \to CD,$$

   které byly aplikovány 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 5.1. 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 takového 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.*

- $Q = \{$

$$\begin{array}{lll}
\langle A,\varepsilon,\varepsilon,\varepsilon\rangle, & \langle A,\varepsilon,b,\varepsilon\rangle, & \langle A,\varepsilon,B,\varepsilon\rangle, \\
\langle C,\varepsilon,\varepsilon,\varepsilon\rangle, & \langle C,\varepsilon,g,\varepsilon\rangle, & \langle C,\varepsilon,G,\varepsilon\rangle, \\
\langle C,\varepsilon,D,DE\to GH\rangle, & \langle E,DE\to GH,\varepsilon,\varepsilon\rangle, & \langle E,DE\to GH,pq,\varepsilon\rangle, \\
\langle E,DE\to GH,pQ,\varepsilon\rangle, & \langle E,DE\to GH,PQ,\varepsilon\rangle, & \langle E,DE\to GH,HG,\varepsilon\rangle, \\
\langle E,DE\to GH,H,\varepsilon\rangle, & \langle E,\varepsilon,E,\varepsilon\rangle, & \mathrm{f}, \ldots
\end{array}$$

  $\}$

- *R obsahuje (mimo dalších zde neuvedených) tato pravidla:*

Označení	***Pravidlo***			***Odp. prav. z G***
I-1	$\langle A,\varepsilon,\varepsilon,\varepsilon\rangle b$	$\to$	$\langle A,\varepsilon,b,\varepsilon\rangle$	—
II-1	$\langle A,\varepsilon,b,\varepsilon\rangle$	$\to$	$\langle A,\varepsilon,B,\varepsilon\rangle$	$B\to b$
IVb-1	$\langle A,\varepsilon,B,\varepsilon\rangle$	$\to$	$\langle C,\varepsilon,\varepsilon,\varepsilon\rangle$	$A\to BC$
I-2	$\langle C,\varepsilon,\varepsilon,\varepsilon\rangle g$	$\to$	$\langle C,\varepsilon,g,\varepsilon\rangle$	—
II-2	$\langle C,\varepsilon,g,\varepsilon\rangle$	$\to$	$\langle C,\varepsilon,G,\varepsilon\rangle$	$G\to g$
IIIa	$\langle C,\varepsilon,G,\varepsilon\rangle$	$\to$	$\langle C,\varepsilon,D,DE\to GH\rangle$	$DE\to GH$
IVb-2	$\langle C,\varepsilon,D,DE\to GH\rangle$	$\to$	$\langle E,DE\to GH,\varepsilon,\varepsilon\rangle$	$C\to DE$
I-3	$\langle E,DE\to GH,\varepsilon,\varepsilon\rangle pq$	$\to$	$\langle E,DE\to GH,pq,\varepsilon\rangle$	—
II-3	$\langle E,DE\to GH,pq,\varepsilon\rangle$	$\to$	$\langle E,DE\to GH,pQ,\varepsilon\rangle$	$Q\to q$
II-4	$\langle E,DE\to GH,pQ,\varepsilon\rangle$	$\to$	$\langle E,DE\to GH,PQ,\varepsilon\rangle$	$P\to p$
IIIc	$\langle E,DE\to GH,PQ,\varepsilon\rangle$	$\to$	$\langle E,DE\to GH,HG,\varepsilon\rangle$	$HG\to PQ$
IVa	$\langle E,DE\to GH,HG,\varepsilon\rangle$	$\to$	$\langle E,DE\to GH,H,\varepsilon\rangle$	$H\to HG$
IIIb	$\langle E,DE\to GH,H,\varepsilon\rangle$	$\to$	$\langle E,\varepsilon,E,\varepsilon\rangle$	$DE\to GH$
V	$\langle E,\varepsilon,E,\varepsilon\rangle$	$\to$	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 $B \to b$*
$\vdash$	$\langle C, \varepsilon, \varepsilon, \varepsilon \rangle gpq$	IVb-1	*ustavení nové hlavy podle $A \to BC$*
$\vdash$	$\langle C, \varepsilon, g, \varepsilon \rangle pq$	I-2	*načtení g ze vstupní pásky*
$\vdash$	$\langle C, \varepsilon, G, \varepsilon \rangle pq$	II-2	*redukce podle $G \to g$*
$\vdash$	$\langle C, \varepsilon, D, DE \to GH \rangle pq$	IIIa	*viz Komentář 1 níže*
$\vdash$	$\langle E, DE \to GH, \varepsilon, \varepsilon \rangle pq$	IVb-2	*ustavení nové hlavy podle $C \to DE$*
$\vdash$	$\langle E, DE \to GH, pq, \varepsilon \rangle$	I-3	*načtení pq ze vstupní pásky*
$\vdash$	$\langle E, DE \to GH, pQ, \varepsilon \rangle$	II-3	*redukce podle $Q \to q$*
$\vdash$	$\langle E, DE \to GH, PQ, \varepsilon \rangle$	II-4	*redukce podle $P \to p$*
$\vdash$	$\langle E, DE \to GH, HG, \varepsilon \rangle$	IIIc	*redukce podle $HG \to PQ$ uvnitř sloupce (!)*
$\vdash$	$\langle E, DE \to GH, H, \varepsilon \rangle$	IVa	*redukce podle $H \to HG$ 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 $DE \to GH$ (redukce G na D) a uložení tohoto pravidla do čtvrté komponenty stavu automatu

Komentář 2: simulace druhé části aplikace kontextového pravidla $DE \to GH$ (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.

Rigorózní důkaz

Nejprve dokážeme inkluzi $L(G) \subseteq L(M)$. Pomocí matematické indukce pro $i \geq 0$ budeme demonstrovat platnost Tvrzení A, B, a C.

Tvrzení A.

$$S \Rightarrow^i \lambda\alpha x\beta\mu$$

v G implikuje

$$f \dashv \langle X, \varepsilon, X, \varepsilon \rangle \dashv^* \langle Z, u, \alpha x\beta, v \rangle \omega$$

v M, kde $X, Z \in N$, $\alpha, \beta, \lambda, \mu \in V^*$, $x \in N^2 \cup N \cup T \cup \{\varepsilon\}$, $\omega \in T^*$.

Základ indukce. Nechť $i = 0$. Pak

$$S \Rightarrow^0 S$$

v G. Podle bodu V konstrukce $\langle X, \varepsilon, X, \varepsilon\rangle \to f \in R$ pro každé $X \in N$, takže

$$f \dashv \langle S, \varepsilon, S, \varepsilon\rangle \dashv^0 \langle S, \varepsilon, S, \varepsilon\rangle$$

v M.

Indukční hypotéza. Předpokládejme, že Tvrzení A platí pro všechna $i \leq n$, kde n je kladné celé číslo.

Indukční krok. Uvažujme libovolnou derivaci tvaru

$$S \Rightarrow^{n+1} \lambda\alpha y\beta\mu$$

a rozepišme ji přesněji jako

$$S \Rightarrow^{n} \lambda\alpha x\beta\mu \Rightarrow \lambda\alpha y\beta\mu,$$

kde $x, y \in N^2 \cup N \cup T \cup \{\varepsilon\}$, $\alpha, \beta, \lambda, \mu \in V^*$.

Podle indukční hypotézy

$$f \dashv^* \langle Z, u, \alpha x\beta, v\rangle\omega \dashv \langle Z, u, \alpha y\beta, v\rangle\omega$$

v M. Následující tři případy představují rozbor všech možností, kterými může G provést derivaci $\lambda\alpha x\beta\mu \Rightarrow \lambda\alpha y\beta\mu$.

a) Nechť $A \to a \in P$, $x = A$, $y = a$, kde $a \in T \cup \{\varepsilon\}$, $A \in N$. Potom

$$\lambda\alpha A\beta\mu \Rightarrow \lambda\alpha a\beta\mu[A \to a]$$

v G. Podle bodu II konstrukce $\langle X, u, \alpha a\beta, v\rangle \to \langle X, u, \alpha A\beta, v\rangle \in R$, takže

$$\langle X, u, \alpha x\beta, v\rangle\omega \dashv \langle X, u, \alpha y\beta, v\rangle\omega$$

v M.

b) Nechť $A \to BC \in P$, $x = A$, $y = BC$, kde $A, B, C \in N$. Pak

$$\lambda\alpha A\beta\mu \Rightarrow \lambda\alpha BC\beta\mu[A \to BC]$$

v G. Podle konstrukce bodu IVa $\langle X, u, \alpha BC\beta, v\rangle \to \langle X, u, \alpha A\beta, v\rangle \in R$, takže

$$\langle X, u, \alpha x\beta, v\rangle\omega \dashv \langle X, u, \alpha y\beta, v\rangle\omega$$

v M.

c) Nechť $AB \to CD \in P$, $x = AB$, $y = CD$, where $A, B, C, D \in N$. Pak,

$$\lambda\alpha AB\beta\mu \Rightarrow \lambda\alpha CD\beta\mu[AB \to CD]$$

v G. Podle konstrukce bodu IIIc $\langle X, u, \alpha CD\beta, v\rangle \to \langle X, u, \alpha AB\beta, v\rangle \in R$, takže

$$\langle X, u, \alpha x\beta, v\rangle\omega \dashv \langle X, u, \alpha y\beta, v\rangle\omega$$

v M.

Tvrzení A tedy platí. $\square$

Tvrzení B resp. C zkoumají kontextové derivace na hranicích resp. ustavení nové hlavy.

Tvrzení B. Nechť x,$y \in T^*$ jsou dva sousední terminální segmenty libovolné věty $w \in L(G)$, $|x|$, $|y| \leq k$. Jestliže v G existuje derivace tvaru

$$\begin{array}{ll} & \alpha_0 x_0 y_0 \beta_0 \\ \Rightarrow^* & \alpha_1 x_1 A_1 B_1 y_1 \beta_1 \\ \Rightarrow & \alpha_1 x_1 C_1 D_1 y_1 \beta_1 [p_1] \\ \Rightarrow^* & \alpha_2 x_2 A_2 B_2 y_2 \beta_2 \\ \Rightarrow & \alpha_2 x_2 C_2 D_2 y_2 \beta_2 [p_2] \\ & \dots \\ \Rightarrow^* & \alpha_i x_i A_i B_i y_i \beta_i \\ \Rightarrow & \alpha_i x_i C_i D_i y_i \beta_i [p_i] \\ \Rightarrow^* & \alpha x y \beta, \end{array}$$

kde $p_i = A_i B_i \to C_i D_i \in P_{CS}$ přepisují A_i a B_i na hranici kontextovým způsobem ($A_j, B_j, C_j, D_j \in N$, $\alpha, \beta, \alpha_m, \beta_m \in V^*$ pro $j = 1, 2, \dots, i$, $m = 0, 1, \dots, i$), potom

$$\begin{array}{ll} & \langle X, u, x, \varepsilon \rangle y\omega \\ \vdash^* & \langle X, u_i, x_i C_i, \varepsilon \rangle y\omega \\ \vdash & \langle X, u_i, x_i A_i, p_i \rangle y\omega \\ & \dots \\ \vdash^* & \langle X, u_2, x_2 C_2, p_i \dots p_3 \rangle y\omega \\ \vdash & \langle X, u_2, x_2 A_2, p_i \dots p_3 p_2 \rangle y\omega \\ \vdash^* & \langle X, u_1, x_1 C_1, p_i \dots p_3 p_2 \rangle y\omega \\ \vdash & \langle X, u_1, x_1 A_1, p_i \dots p_3 p_2 p_1 \rangle y\omega \\ \vdash^* & \langle X, u_0, x_0, p_i \dots p_3 p_2 p_1 \rangle y\omega \end{array}$$

a zároveň

$$\begin{array}{ll} & \langle Y, p_i p_{i-1} \dots p_2 p_1, y, \varepsilon \rangle \omega \\ \vdash^* & \langle Y, p_i p_{i-1} \dots p_2 p_1, D_i y_i, v_i \rangle \omega \\ \vdash & \langle Y, p_{i-1} \dots p_2 p_1, B_i y_i, v_i \rangle \omega \\ & \dots \\ \vdash^* & \langle Y, p_2 p_1, D_2 y_2, v_2 \rangle \omega \\ \vdash & \langle Y, p_1, B_2 y_2, v_2 \rangle \omega \\ \vdash^* & \langle Y, p_1, D_1 y_1, v_1 \rangle \omega \\ \vdash & \langle Y, \varepsilon, B_1 y_1, v_1 \rangle \omega \\ \vdash^* & \langle Y, \varepsilon, y_0, v_0 \rangle \omega \end{array}$$

v M, kde $x_j, y_j \in V^*$, $u, v, u_j, v_j \in P_{CS}^*$, $|u|, |v|, |u_j|, |v_j| \leq l$, $0 \leq j \leq i$, $i \leq l$, $\omega \in T^*$.

Základ indukce. Nechť $i = 0$. Pak

$$\alpha_0 x_0 y_0 \beta_0 \Rightarrow^* \alpha x y \beta$$

v G. V tomto případě se nevyskytuje žádná kontextová derivace na hranici mezi vyšetřovanými sloupci. Pravidla jsou tedy aplikována pouze uvnitř těchto sloupců. V M tudíž

$$\langle X, u, x, \varepsilon \rangle y\omega \vdash^* \langle X, u_0, x_0, \varepsilon \rangle y\omega \quad \text{a} \quad \langle Y, \varepsilon, y, \varepsilon \rangle \omega \vdash^* \langle Y, \varepsilon, y_0, v_0 \rangle \omega,$$

kde jsou simulovány pouze redukce uvnitř sloupců (viz Tvrzení A).

Indukční hypotéza. Předpokládejme, že Tvrzení B platí pro každé $0 \leq i \leq t$ pro nějaké $t = \{0, 1, 2, \ldots, l\}$.

Indukční krok. Uvažujme dva sousední terminální segmenty x a y ($x, y \in T^*$) v tomto pořadí a libovolnou derivaci tvaru

$$
\begin{array}{ll}
 & \alpha_0 x_0 y_0 \beta_0 \\
\Rightarrow^* & \alpha_1 x_1 A_1 B_1 y_1 \beta_1 \\
\Rightarrow & \alpha_1 x_1 C_1 D_1 y_1 \beta_1 [p_1] \\
\Rightarrow^* & \alpha_2 x_2 A_2 B_2 y_2 \beta_2 \\
\Rightarrow & \alpha_2 x_2 C_2 D_2 y_2 \beta_2 [p_2] \\
 & \ldots \\
\Rightarrow^* & \alpha_t x_t A_t B_t y_t \beta_t \\
\Rightarrow & \alpha_t x_t C_t D_t y_t \beta_t [p_t] \\
\Rightarrow^* & \alpha_{t+1} x_{t+1} A_{t+1} B_{t+1} y_{t+1} \beta_{t+1} \\
\Rightarrow & \alpha_{t+1} x_{t+1} C_{t+1} D_{t+1} y_{t+1} \beta_{t+1} [p_{t+1}] \\
\Rightarrow^* & \alpha x y \beta
\end{array}
$$

v G.

Podle indukční hypotézy

$$
\begin{array}{ll}
 & \langle X, u, x, \varepsilon \rangle y\omega \\
\vdash^* & \langle X, u_{t+1}, x_{t+1} C_{t+1}, \varepsilon \rangle y\omega \\
\vdash & \langle X, u_{t+1}, x_{t+1} A_{t+1}, p_{t+1} \rangle y\omega \\
\vdash^* & \langle X, u_t, x_t C_t, p_{t+1} \rangle y\omega \\
\vdash & \langle X, u_t, x_t A_t, p_{t+1} p_t \rangle y\omega \\
 & \ldots \\
\vdash^* & \langle X, u_2, x_2 C_2, p_{t+1} p_t \ldots p_3 \rangle y\omega \\
\vdash & \langle X, u_2, x_2 A_2, p_{t+1} p_t \ldots p_3 p_2 \rangle y\omega \\
\vdash^* & \langle X, u_1, x_1 C_1, p_{t+1} p_t \ldots p_3 p_2 \rangle y\omega \\
\vdash & \langle X, u_1, x_1 A_1, p_{t+1} p_t \ldots p_3 p_2 p_1 \rangle y\omega \\
\vdash^* & \langle X, u_0, x_0, p_{t+1} p_t \ldots p_3 p_2 p_1 \rangle y\omega
\end{array}
$$

a zároveň

$$
\begin{array}{ll}
 & \langle Y, p_{t+1} p_t p_{t-1} \ldots p_2 p_1, y, \varepsilon \rangle \omega \\
\vdash^* & \langle Y, p_{t+1} p_t p_{t-1} \ldots p_2 p_1, D_{t+1} y_{t+1}, v_{t+1} \rangle \omega \\
\vdash & \langle Y, p_t p_{t-1} \ldots p_2 p_1, B_{t+1} y_{t+1}, v_{t+1} \rangle \omega \\
\vdash^* & \langle Y, p_t p_{t-1} \ldots p_2 p_1, D_t y_t, v_t \rangle \omega \\
\vdash & \langle Y, p_{t-1} \ldots p_2 p_1, B_t y_t, v_t \rangle \omega \\
 & \ldots \\
\vdash^* & \langle Y, p_2 p_1, D_2 y_2, v_2 \rangle \omega \\
\vdash & \langle Y, p_1, B_2 y_2, v_2 \rangle \omega \\
\vdash^* & \langle Y, p_1, D_1 y_1, v_1 \rangle \omega \\
\vdash & \langle Y, \varepsilon, B_1 y_1, v_1 \rangle \omega \\
\vdash^* & \langle Y, \varepsilon, y_0, v_0 \rangle \omega
\end{array}
$$

v M. Z bodů IIIa a IIIb konstrukce vyplývá, že pro každé pravidlo tvaru $AB \to CD \in P$ existují v R pravidla tvaru

$$\langle X, u, \alpha C, v\rangle \rightarrow \langle X, u, \alpha A, vAB \rightarrow CD\rangle \quad \text{a} \quad \langle X, AB \rightarrow CDu, D\beta, v\rangle \rightarrow \langle X, u, B\beta, v\rangle.$$

Tato pravidla simulují přechody tvaru

$$\langle X, u_r, x_r C_r, p_{t+1} \dots p_{r-1}\rangle y\omega \vdash \langle X, u_r, x_r A_r, p_{t+1} \dots p_{r-1} p_r\rangle y\omega$$

a

$$\langle Y, p_s p_{s-1} \dots p_1, D_s y_s, v_s\rangle \omega \vdash \langle Y, p_{s-1} \dots p_1, B_s y_s, v_s\rangle \omega$$

v M, kde $r, s = t+1, t, \dots, 1$, takže Tvrzení B platí.

Všechny ostatní derivační kroky a přechody v G a M (označené ve výše uvedených sekvencích symboly $\Rightarrow^*$ and $\vdash^*$) jsou zkoumány v Tvrzeních A a C. $\square$

Tvrzení C.

$$\lambda z\mu \Rightarrow^i \lambda' x\mu'$$

v G implikuje

$$\langle Z, u, \alpha z\beta, v\rangle \omega \vdash^* \langle A, u', Y, v'\rangle \omega'$$

v M, kde $\lambda, \lambda', \mu, \mu' \in V^*$, $z, x \in N^2 \cup N$, $A, Z \in N$, $\omega, \omega' \in T^*$. Pro $|x| = 1$ jsou $A = x$, $u' = \varepsilon$, $Y \in N$, $v' \in P_{CS}^*$ a pro $x = x_1 x_2$, kde $x_1, x_2 \in N$, jsou $A = x_2$, $u' \in P_{CS}^*$, $Y, v' = \varepsilon$.

Základ indukce. Nechť $i = 0$. Pak

$$\lambda z\mu \Rightarrow^0 \lambda z\mu$$

v G. Rozhodně také

$$\langle Z, u, \alpha z\beta, X, v\rangle \omega \vdash^0 \langle Z, u, \alpha z\beta, X, v\rangle \omega$$

v M.

Indukční hypotéza. Předpokládejme, že implikace v Tvrzení C platí pro každé $i \leq g$, kde g je kladné celé číslo.

Indukční krok. Uvažujme libovolnou derivaci tvaru

$$\lambda z\mu \Rightarrow^{g+1} \lambda' y\mu'$$

a vyjádřeme ji jako

$$\lambda z\mu \Rightarrow^g \lambda' x\mu' \Rightarrow \lambda' y\mu',$$

kde $x, y, z \in N^2 \cup N$, $\lambda, \mu, \lambda', \mu' \in V^*$.

Podle indukční hypotézy

$$\langle A, u, Y, v\rangle \omega \vdash \langle A', u', Y', v'\rangle \omega'$$

v M. Dále prozkoumáme derivaci

$$\lambda x\mu \Rightarrow \lambda y\mu$$

v G.

Nechť $A \rightarrow BC \in P$, $x = A$, $y = y_1 y_2 = BC$, kde $A, B, C \in N$. Pak

$$\lambda A\mu \Rightarrow \lambda BC\mu[A \rightarrow BC]$$

v G. Podle konstrukce bodu IVb $\langle A, \varepsilon, B, v\rangle \to \langle C, v, \varepsilon, \varepsilon\rangle \in R$, takže

$$\langle x, \varepsilon, y_1, v\rangle\omega \vdash \langle y_2, v, \varepsilon, \varepsilon\rangle\omega$$

v M a Tvrzení C platí. $\square$

Celkově Tvrzení A, B a C dokazují, že $L(G) \subseteq L(M)$.

Abychom dokázali i obrácenou inkluzi, tedy $L(M) \subseteq L(G)$, budeme demonstrovat platnost Tvrzení D, E, F a G. Tvrzení D popisuje tvar každé úspěšné výpočetní sekvence v M. V Tvrzeních E, F a G se zaměříme podrobněji na jednotlivé části této sekvence.

Tvrzení D. M přijímá každou větu $z \in L(M)$ sekvencí přechodů tvaru

$$\begin{array}{lll}
 & \langle S, \varepsilon, \varepsilon, \varepsilon\rangle z_1 \dots z_m & \\
\vdash & \langle S, \varepsilon, z_1, \varepsilon\rangle z_2 \dots z_m & [p_1] \\
\vdash^* & \langle X_1, \varepsilon, Y_1, v_1\rangle z_2 \dots z_m & [\rho_1] \\
\vdash & \langle X_2, v_1, \varepsilon, \varepsilon\rangle z_2 \dots z_m & [h_1] \\
\vdash & \langle X_2, v_1, z_2, \varepsilon\rangle z_3 \dots z_m & [p_2] \\
\vdash^* & \langle X_2, \varepsilon, Y_2, v_2\rangle z_3 \dots z_m & [\rho_2] \\
\vdash & \langle X_3, v_2, \varepsilon, \varepsilon\rangle z_3 \dots z_m & [h_2] \\
\vdash & \langle X_3, v_2, z_3, \varepsilon\rangle z_4 \dots z_m & [p_3] \\
 & \dots & \\
\vdash^* & \langle X_{m-1}, \varepsilon, Y_{m-1}, v_{m-1}\rangle z_m & [\rho_{m-1}] \\
\vdash & \langle X_m, v_{m-1}, \varepsilon, \varepsilon\rangle z_m & [h_{m-1}] \\
\vdash & \langle X_m, v_{m-1}, z_m, \varepsilon\rangle & [p_m] \\
\vdash^* & \langle X_m, \varepsilon, X_m, \varepsilon\rangle & [\rho_m] \\
\vdash & f, &
\end{array}$$

kde z je tvaru $z = z_1 \dots z_m$, $m \geq 0$, $z_i \in T^*$ pro $i = 1, \dots, m$, $|z_i| \leq k$, $p_1, \dots, p_m \in R_I$, $\rho_1 \in (R_{IN} \cup R_{CS1})^*$, $\rho_m \in (R_{IN} \cup R_{CS2})^*$, $\rho_2, \dots, \rho_{m-1} \in (R_{IN} \cup R_{CS1} \cup R_{CS2})^*$, a $h_1, \dots, h_{m-1} \in R_{HEAD}$, $f \in F$.

Rozeberme nyní podrobněji jednotlivé části.

M začíná ve výchozí konfiguraci $\langle S, \varepsilon, \varepsilon, \varepsilon\rangle z_1 \dots z_m$. Použitím některého pravidla z R_I načte první terminální segment z_1, kde $|z_1| \leq k$. Podle konstrukce je použité pravidlo p_1 nutně tvaru $\langle S, \varepsilon, \varepsilon, \varepsilon\rangle z_1 \to \langle S, \varepsilon, z_1, \varepsilon\rangle \in R_I$.

V další výpočetní sekvenci jsou použita pravidla z $R_{IN} \cup R_{CS1}$, která simulují redukce prvního načteného segmentu. Každé pravidlo z R_{CS1} použité při výpočtu simuluje první část aplikace kontextového pravidla z P na hranici prvního a druhého sloupce. Tato simulovaná pravidla jsou zaznamenávána ve čtvrté komponentě stavu automatu. Zopakujme, že počet aplikací pravidel z R_{CS1} ve sloupci je limitován hodnotou l. Jakmile je z_1 redukován na jediný nonterminál, automat ustaví novou hlavu použitím vhodného pravidla z R_{HEAD}. Tím je vyprázdněna druhá a třetí komponenta stavu automatu a zaznamenaná pravidla odsimulovaná pomocí pravidel z R_{CS1} jsou přesunuta do druhé komponenty.

M načte další terminální segment použitím některého pravidla z R_I.

V následující části M redukuje načtený segment pomocí pravidel z $R_{IN} \cup R_{CS1} \cup R_{CS2}$, kde pravidla z R_{CS2} kontrolují aplikace pravidel z R_{CS1} v předchozím sloupci a provádějí druhou část simulace aplikace kontextových pravidel ve stejném pořadí jako v předchozím sloupci. Kromě toho je možné použít pravidla z R_{CS1} k simulaci aplikací kontextových pravidel na hranici s následujícím sloupcem. Ostatní pravidla z R_{IN} simulují redukce uvnitř sloupce. Jakmile je druhá komponenta prázdná a načtený segment je zredukován na jediný nonterminál, M ustaví novou hlavu, přesune obsah čtvrté komponenty do druhé komponenty, vyprázdní třetí a čtvrtou komponentu a načte další terminální segment.

Tento tvar výpočtu se opakuje přes všechny terminální segmenty až do předposledního terminálního segmentu z_{m-1}.

Jakmile M načte poslední terminální segment z_m, používá pak pravidla z $R_{IN} \cup R_{CS2}$, aby zredukoval z_m na jediný nonterminál. Poznamenejme, že v této části jsou simulovány pouze redukce uvnitř sloupce a pomocí pravidel z R_{CS2} jsou kontrolovány aplikace pravidel z R_{CS1} z předchozího sloupce. Poté co je z_m zredukován na jediný nonterminál shodný s nonterminálem v první komponentě stavu a druhá a třetí komponenta jsou prázdné, M přechází do koncového stavu pomocí některého pravidla z R_F. Tímto způsobem je věta z přijata automatem M. $\square$

Tvrzení E. Jestliže ρ_i (viz Tvrzení D) je tvaru

$$\alpha_1 s_1 \alpha_2 s_2 \dots \alpha_n s_n \alpha_{n+1},$$

pak ρ_{i+1} je tvaru

$$\beta_1 t_1 \beta_2 t_2 \dots \beta_n t_n \beta_{n+1},$$

kde $\alpha_1, \dots, \alpha_{n+1} \in (R_{CS2} \cup R_{IN})^*$, $s_1, \dots, s_n \in R_{CS1}$, $\beta_1, \dots, \beta_{n+1} \in (R_{CS1} \cup R_{IN})^*$, $t_1, \dots, t_n \in R_{CS2}$, $n \leq l$, $1 \leq i < m$.

Všimněme si, že podle konstrukce je s_j tvaru

$$\begin{array}{ll} & \langle X, u, \gamma C_j, A_1 B_1 \to C_1 D_1 \dots A_{j-1} B_{j-1} \to C_{j-1} D_{j-1} \rangle \\ \to & \langle X, u, \gamma A_j, A_1 B_1 \to C_1 D_1 \dots A_j B_j \to C_j D_j \rangle, \end{array}$$

kde $1 < j \leq n$. Pro $j = 1$ je s_1 tvaru

$$\langle X, u, \gamma C_1, \varepsilon \rangle \to \langle X, u, \gamma A_1, A_1 B_1 \to C_1 D_1 \rangle.$$

V dalším sloupci je t_j tvaru

$$\begin{array}{ll} & \langle Y, A_j B_j \to C_j D_j \dots A_n B_n \to C_n D_n, D_j \delta, v \rangle \\ \to & \langle Y, A_{j+1} B_{j+1} \to C_{j+1} D_{j+1} \dots A_n B_n \to C_n D_n, B_j \delta, v \rangle, \end{array}$$

kde $1 \leq j < n$. Pro $j = n$ je t_n tvaru

$$\langle Y, A_n B_n \to C_n D_n, D_n \delta, v \rangle \to \langle Y, \varepsilon, B_n \delta, v \rangle,$$

kde $A_j B_j \to C_j D_j \in P$, pro $1 \leq j \leq n$.

Pravidla s_j a t_j odpovídají pravidlu $A_j B_j \to C_j D_j \in P$ pro $j = 1, \dots, n$. Pravidlo s_j simuluje redukci C_j na A_j a přidává $A_j B_j \to C_j D_j$ na konec čtvrté komponenty stavu automatu. V dalším sloupci simuluje t_j redukci D_j na B_j a odebírá $A_j B_j \to C_j D_j$ z druhé komponenty, ve které je $A_j B_j \to C_j D_j$ nejlevějším pravidlem.

Korespondence s_j a t_j a shodné pořadí jejich aplikací jsou velmi důležité. Tím automat vyprázdní druhou komponentu svého stavu. Když je navíc zredukován aktuální segment na jediný nonterminál, může být ustavena nová hlava a načten další terminální segment. $\square$

Tvrzení F. Jestliže je přechod proveden pomocí některého pravidla $h_i \in R_{HEAD}$, pak $p_{i+1} \in R_I$ je následující použité pravidlo. Poznamenejme, že některá symbolika v této části vychází z Tvrzení D.

Podle konstrukce je jediná možnost pro ustavení nové hlavy C stav automatu tvaru

$$\langle A, \varepsilon, B, v\rangle,$$

kde $A, B, C \in N$, $v \in P^*_{CS}$, $|v| \leq l$, a $A \to BC \in P$. Stav $\langle A, \varepsilon, B, v\rangle$ může být dosažen pouze korektní aplikací ρ_i (druhá komponenta je prázdná, viz Tvrzení E). Poté je pomocí pravidla tvaru $\langle A, \varepsilon, B, v\rangle \to \langle C, v, \varepsilon, \varepsilon\rangle$ ustavena nová hlava. Po jeho aplikaci se automat nachází ve stavu $\langle C, v, \varepsilon, \varepsilon\rangle$. Všimněme si, že podle konstrukce je jediné pravidlo, které může být v tomto místě použito, $p_{i+1} \in R_I$. Jeho aplikací je načten další terminální segment z_{i+1} a automat poté simuluje redukce tohoto segmentu použitím sekvence pravidel ρ_{i+1}. $\square$

Tvrzení G. Jestliže je ρ_i (viz Tvrzení D) tvaru

$$\gamma_1 r_1 \gamma_2 r_2 \cdots \gamma_o r_o \gamma_{o+1},$$

kde $\gamma_1, \ldots, \gamma_{o+1} \in (R_{CS1} \cup R_{CS2})^*$, $r_1, \ldots, r_o \in R_{IN}$, potom každé pravidlo r_j, kde $1 \leq j \leq o$, simuluje redukce uvnitř sloupce.

Každé r_j může mít jeden z následujících tvarů:

1. $\langle X, u, \alpha a \beta, v\rangle \to \langle X, u, \alpha A \beta, v\rangle$ pro $A \to a \in P$
2. $\langle X, u, \alpha BC \beta, v\rangle \to \langle X, u, \alpha A \beta, v\rangle$ pro $A \to BC \in P$
3. $\langle X, u, \alpha CD \beta, v\rangle \to \langle X, u, \alpha AB \beta, v\rangle$ pro $AB \to CD \in P$

kde $A, B, C, D \in N$, $a \in T \cup \{\varepsilon\}$, $\alpha, \beta \in V^*$, $u, v \in P^*_{CS}$, $|u|, |v| \leq l$. Tato pravidla ovlivňují pouze třetí komponentu stavu automatu, kde jsou simulovány vnitřní redukce aktuálního segmentu. $\square$

Z výše uvedených tvrzení vyplývá, že

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

v G, kde $x_{n1} = z_1$, $x_{n2} = z_2$, $\ldots$, $x_{nm} = z_m$.

Tvrzení A až G dokazují, že $L(M) = L(G)$. Tím je důkaz hotový. $\blacksquare$

5.2 Shrnutí

Uvedená kapitola představuje zcela nový pohled na derivační proces v obecných gramatikách. Protože tyto gramatiky mají shodnou sílu s Turingovým strojem, bylo od počátku jasné, že zvýšit jejich sílu již není možné.

Síla Kurodovy normální formy však mohla být zachována, případně snížena jen částečně. Prezentovaný výsledek je do jisté míry překvapivý, neboť generativní schopností obecných gramatik byly zavedenými omezeními radikálně sníženy až na úroveň regulárních jazyků. Podíváme-li se však blíže na některé modifikace bezkontextových gramatik (maticové gramatiky, programované gramatiky, gramatiky s náhodným kontextem), mají zavedena rovněž určitá omezení kladená na aplikace jednotlivých přepisovacích pravidel pro aktuální větnou formu, případně s ohledem na dříve aplikovaná pravidla. Jejich generativní síla však byla těmito omezeními naopak zvýšena. Toto dále umocňuje hodnotu námi dosaženého výsledku.

Kapitola 6

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 [5] a v [8], 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.

6.1 Základní definice

Definice 6.1 — bezkontextová gramatika nad volnou grupou s redukovaným počtem nonterminálů *Bezkontextová gramatika nad volnou grupou s redukovaným počtem nonterminálů* (dále jen **CF°R** gramatika podle anglického označení *context-free—reduced*, kde ° 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

$$A \rightarrow \alpha,$$

  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 6.2 — 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 6.3 — 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 6.4 — 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 6.5 — 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 $\circ\!\!\Rightarrow^*$.

6.2 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 6.1 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 6.2 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

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

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

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

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 $\boldsymbol{CF^{\circ}R}$ *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 6.1.*

Nyní zkonstruujeme $\boldsymbol{CF^{\circ}R}$ *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	$S_\Gamma \to 01111110\mathbf{2}$	—
IIa	$2 \to \overline{0001100020011111002}\mathbf{2}01111110\mathbf{2}00011000\mathbf{2}$	$GB \to SB$
IIb	$2 \to \overline{000110002101001012}\mathbf{2}00100100\mathbf{2}10100101\mathbf{2}$	$ZB \to CZ$
IIc	$2 \to \overline{101001012001001002}\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	$CZ \to BZ$
IId	$2 \to \overline{000110002100110012}\mathbf{2}10011001\mathbf{2}10011001\mathbf{2}$	$YB \to YY$
IIIa	$2 \to \overline{011111102}\mathbf{2}10000001\mathbf{2}01000010\mathbf{2}$	$S \to XP$
IIIb	$2 \to \overline{010000102}\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}$	$P \to YZ$
IIIc	$2 \to \overline{011111102}\mathbf{2}10000001\mathbf{2}01011010\mathbf{2}$	$S \to XQ$
IIId	$2 \to \overline{010110102}\mathbf{2}00111100\mathbf{2}01100110\mathbf{2}$	$Q \to GR$
IIIe	$2 \to \overline{011001102}\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}$	$R \to BZ$
IVa	$2 \to \overline{10000001}x$	$X \to x$
IVb	$2 \to \overline{10011001}y$	$Y \to y$
IVc	$2 \to \overline{10100101}z$	$Z \to z$

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

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

Nyní vygenerujeme stejnou větu dříve zkonstruovanou $\boldsymbol{CF^{\circ}R}$ *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.*

$$
\begin{array}{lll}
 & S_\Gamma & \\
\Rightarrow & 01111110\mathbf{2} & \text{Ia} \\
\Rightarrow & 01111110\overline{011111102}\mathbf{2}10000001\mathbf{2}01011010\mathbf{2} & \text{IIIc} \\
= & 10000001\mathbf{2}01011010\mathbf{2} & \text{redukce} \\
\Rightarrow & 10000001\mathbf{2}01011010\overline{010110102}\mathbf{2}00111100\mathbf{2}01100110\mathbf{2} & \text{IIId} \\
= & 10000001\mathbf{2}00111100\mathbf{2}01100110\mathbf{2} & \text{redukce} \\
\Rightarrow & 10000001\mathbf{2}00111100\mathbf{2}01100110\overline{011001102}\mathbf{2}00011000\mathbf{2}10100101\mathbf{2} & \text{IIIe} \\
= & 10000001\mathbf{2}00111100\mathbf{2}00011000\mathbf{2}10100101\mathbf{2} & \text{redukce} \\
\Rightarrow & 10000001\mathbf{2}00111100\mathbf{2}00011000\overline{000110002001111002}\mathbf{2}01111110\mathbf{2} & \\
 & 00011000\mathbf{2}10100101\mathbf{2} & \text{IIa} \\
= & 10000001\mathbf{2}01111110\mathbf{2}00011000\mathbf{2}10100101\mathbf{2} & \text{redukce} \\
\Rightarrow & 10000001\mathbf{2}01111110\overline{011111102}\mathbf{2}10000001\mathbf{2}01000010\mathbf{2}00011000\mathbf{2} & \\
 & 10100101\mathbf{2} & \text{IIIa} \\
= & 10000001\mathbf{2}10000001\mathbf{2}01000010\mathbf{2}00011000\mathbf{2}10100101\mathbf{2} & \text{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} & \text{IIIb} \\
= & 10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}00011000\mathbf{2}10100101\mathbf{2} & \text{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} & \text{IIb} \\
= & 10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}00100100\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{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} & \text{IIc} \\
= & 10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}00011000\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{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} & \text{IId} \\
= & 10000001\mathbf{2}10000001\mathbf{2}10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{redukce} \\
\Rightarrow & 10000001\overline{10000001}x10000001\mathbf{2}10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{IVa} \\
= & x10000001\mathbf{2}10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{redukce} \\
\Rightarrow & x10000001\overline{10000001}x10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{IVa} \\
= & xx10011001\mathbf{2}10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{redukce} \\
\Rightarrow & xx10011001\overline{10011001}y10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{IVb} \\
= & xxy10011001\mathbf{2}10100101\mathbf{2}10100101\mathbf{2} & \text{redukce} \\
\Rightarrow & xxy10011001\overline{10011001}y10100101\mathbf{2}10100101\mathbf{2} & \text{IVb} \\
= & xxyy10100101\mathbf{2}10100101\mathbf{2} & \text{redukce} \\
\Rightarrow & xxyy10100101\overline{10100101}z10100101\mathbf{2} & \text{IVc} \\
= & xxyyz10100101\mathbf{2} & \text{redukce} \\
\Rightarrow & xxyyz10100101\overline{10100101}z & \text{IVc} \\
= & xxyyzz & \text{redukce}
\end{array}
$$

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 $\mathbf{2}$*, 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.*

Rigorózní důkaz

Nejprve dokážeme inkluzi $L(G) \subseteq L(\Gamma)^{\circ}$. Pomocí matematické indukce budeme demonstrovat Tvrzení A a B. Bez ztráty obecnosti předpokládejme, že každá věta $w \in L(G)$ může být generována derivací tvaru

$$S \Rightarrow^{*} w' \Rightarrow^{*} w,$$

kde $w' \in N^{*}$ je generováno z S pouze pomocí pravidel tvaru $AB \to CD$, $A \to BC$ a $A \to \varepsilon$, zatímco výsledná věta w je z w' získána pouze pomocí pravidel tvaru $A \to a$, kde $A, B, C, D \in N$ a $a \in T$.

Tvrzení A. Jestliže

$$S \Rightarrow^{i} y_1 y_2 \ldots y_m$$

v G, potom

$$S_{\Gamma} \circ\!\!\Rightarrow^{i+1} h(y_1)2h(y_2)2 \ldots h(y_m)2$$

v Γ, kde $y_1, \ldots, y_m \in N$, $m \geq 0$.

Základ indukce. Nechť $i = 0$. Pak $S \Rightarrow^{0} S$ v G. Podle konstrukce $S_{\Gamma} \to h(S)2 \in P_{\Gamma}$, takže $S_{\Gamma} \circ\!\!\Rightarrow^{1} h(S)2$ v Γ.

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

Indukční krok. Uvažujme libovolnou derivaci tvaru

$$S \Rightarrow^{l+1} \beta,$$

kde $\beta \in N^{*}$ a vyjádřeme tuto derivaci jako

$$S \Rightarrow^{l} \alpha \Rightarrow \beta,$$

kde $\alpha \in N^{*}$.

Dále vyjádřeme řetězec β jako $\beta = y_1 y_2 \ldots y_k$, kde $y_1, \ldots, y_k \in N$, $k \geq 0$. Podle indukční hypotézy

$$S_{\Gamma} \circ\!\!\Rightarrow^{l+2} h(y_1)2h(y_2)2 \ldots h(y_k)2 = \beta_{\Gamma}$$

v Γ. Poznamenejme, že v tomto důkazu budeme dále vyjadřovat prefix resp. sufix aktuální větné formy jako $u = p_1 \ldots p_r$ resp. $v = q_1 \ldots q_s$, kde $p_j, q_k \in N$ pro $j = 1, \ldots, r$, $k = 1, \ldots, s$, $r, s \geq 0$. Pro názornost také ve větných formách inverzní podřetězce, které jsou redukovány pomocí volné grupy, podtrhneme. Následující případy od 1 do 3 zkoumají všechny možnosti, kterými může G provést derivaci $\alpha \Rightarrow \beta$.

1. Nechť $AB \to CD \in P$, kde $A, B, C, D \in N$. Potom

   $$\alpha = uABv \Rightarrow uCDv = \beta$$

   v G. Podle bodu II konstrukce $2 \to \overline{h}(B)2\overline{h}(A)\overline{2}2h(C)2h(D)2 \in P_{\Gamma}$ a

   $$\alpha_{\Gamma} = h(p_1)2 \ldots h(p_r)2h(A)2h(B)2h(q_1)2 \ldots h(q_s)2 \circ\!\!\Rightarrow h(p_1)2 \ldots$$

   $$\ldots h(p_r)\underline{2h(A)2h(B)\overline{h}(B)\overline{2}\overline{h}(A)\overline{2}}2h(C)2h(D)2h(q_1)2 \ldots h(q_s)2 =$$

$$= h(p_1)2 \dots h(p_r)2h(C)2h(D)2h(q_1)2 \dots h(q_s)2 = \beta_\Gamma$$

v Γ. Rozhodně tedy

$$S \Rightarrow^{l+1} \beta$$

v G a

$$S_\Gamma \circ\!\!\Rightarrow^{l+2} \beta_\Gamma$$

v Γ.

2. Nechť $A \to BC \in P$, kde $A, B, C \in N$. Pak

$$\alpha = uAv \Rightarrow uBCv = \beta$$

v G. Podle bodu III konstrukce $2 \to \overline{h}(A)\overline{2}2h(B)2h(C)2 \in P_\Gamma$ a

$$\alpha_\Gamma = h(p_1)2 \dots h(p_r)2h(A)2h(q_1)2 \dots h(q_s)2 \circ\!\!\Rightarrow h(p_1)2 \dots$$
$$\dots h(p_r)\underline{2h(A)\overline{h}(A)\overline{2}}2h(B)2h(C)2h(q_1)2 \dots h(q_s)2 =$$
$$= h(p_1)2 \dots h(p_r)2h(B)2h(C)2h(q_1)2 \dots h(q_s)2 = \beta_\Gamma$$

v Γ a tedy i

$$S \Rightarrow^{l+1} \beta$$

v G a

$$S_\Gamma \circ\!\!\Rightarrow^{l+2} \beta_\Gamma$$

v Γ.

3. Nechť $A \to \varepsilon \in P$, kde $A \in N$. Potom

$$\alpha = uAv \Rightarrow uv = \beta$$

v G. Podle bodu IV konstrukce $2 \to \overline{h}(A) \in P_\Gamma$ a

$$\alpha_\Gamma = h(p_1)2 \dots h(p_r)2h(A)2h(q_1)2 \dots h(q_s)2 \circ\!\!\Rightarrow h(p_1)2 \dots$$
$$\dots h(p_r)\underline{2h(A)\overline{h}(A)\overline{2}}2h(q_1)2 \dots h(q_s)2 =$$
$$= h(p_1)2 \dots h(p_r)2h(q_1)2 \dots h(q_s)2 = \beta_\Gamma$$

v Γ. Určitě pak také

$$S \Rightarrow^{l+1} \beta$$

v G a

$$S_\Gamma \circ\!\!\Rightarrow^{l+2} \beta_\Gamma$$

v Γ.

Tím je indukční krok hotový a Tvrzení A tedy platí. □

Tvrzení B. Jestliže

$$y_1 \dots y_k \Rightarrow^k w$$

v G použitím pouze pravidel tvaru $A \to a \in P$, potom

$$h(y_1)2 \dots h(y_k)2 \circ\!\!\Rightarrow^k w$$

v Γ, kde $A, y_i \in N$ pro $i = 1, 2, \ldots, k$, $k \geq 0$, $a \in T$ a $w \in T^*$.

Bez ztráty obecnosti předpokládejme, že w je generováno nejlevější derivací.

Základ indukce. Nechť $k = 0$. Pak $\varepsilon \Rightarrow^0 \varepsilon$ v G. Rozhodně i $\varepsilon \circ\!\!\Rightarrow^0 \varepsilon$ v Γ.

Indukční hypotéza. Předpokládejme, že Tvrzení B platí pro každé $k \leq l$, kde l je kladné celé číslo.

Indukční krok. Uvažujme libovolnou větnou formu $y_1 y_2 \ldots y_l y_{l+1}$, kde $y_i \in N$ pro $i = 1, 2, \ldots, l+1$ a vyjádřeme derivaci z Tvrzení B jako

$$y_1 y_2 \ldots y_l y_{l+1} \Rightarrow^l w y_{l+1} \Rightarrow wa,$$

kde $w \in T^*$, $a \in T$.

Nechť $y_{l+1} \to a \in P$, kde $y_{l+1} \in N$ a $a \in T$. Potom

$$w y_{l+1} \Rightarrow wa$$

v G. Podle bodu IV konstrukce $2 \to \overline{h}(y_{l+1})a \in P_\Gamma$, takže

$$wh(y_{l+1})2 \circ\!\!\Rightarrow w\underline{h(y_{l+1})\overline{h}(y_{l+1})}a = wa$$

v Γ. Rozhodně také

$$h(y_1)2h(y_2)2 \ldots h(y_l)2h(y_{l+1})2 \circ\!\!\Rightarrow^{l+1} wa$$

v Γ.

Tím je důkaz Tvrzení B hotový. $\square$

Tvrzení A a B dokazují platnost inkluze $L(G) \subseteq L(\Gamma)^\circ$.

Nyní dokážeme opačnou inkluzi, tedy $L(\Gamma)^\circ \subseteq L(G)$. Podobně jako v předchozím odstavci budeme předpokládat, že každá věta $w \in L(\Gamma)^\circ$ může být generována derivací tvaru

$$S_\Gamma \circ\!\!\Rightarrow^* w' \circ\!\!\Rightarrow^* w,$$

kde w' je generováno z S_Γ pouze použitím pravidel následujících tvarů:

- $S_\Gamma \to h(S)2$
- $2 \to \overline{h}(B)\overline{2}\overline{h}(A)\overline{2}2h(C)2h(D)2$
- $2 \to \overline{h}(A)\overline{2}2h(B)2h(C)2$
- $2 \to \overline{h}(A)$

Věta w je pak z w' vygenerována pouze pomocí pravidel tvaru

$$2 \to \overline{h}(A)a,$$

kde $A, B, C, D \in N$ a $a \in T$. Poznamenejme, že w' je tvaru $h(y_1)2 \ldots h(y_m)2$, kde $y_j \in N$ pro $j = 1, \ldots, m$, $m \geq 0$. Pomocí matematické indukce pro libovolné $i \geq 0$ budeme nejprve demonstrovat Tvrzení C.

Tvrzení C. Jestliže

$$S_\Gamma \circ\!\!\Rightarrow h(S)2 \circ\!\!\Rightarrow^i h(y_1)2h(y_2)2\ldots h(y_m)2$$

v Γ, potom

$$S \Rightarrow^i y_1y_2\ldots y_m$$

v G, kde $y_j \in N$ pro $j = 1, 2, \ldots, m$.

Základ indukce. Nechť $i = 0$. Pak $m = 1$, $h(y_1) = h(S)$ a

$$S_\Gamma \circ\!\!\Rightarrow h(S)2 \circ\!\!\Rightarrow^0 h(S)2$$

v Γ. V G je $y_1 = S$ a $S \Rightarrow^0 S$.

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

Indukční krok. Uvažujme libovolnou derivaci tvaru

$$S_\Gamma \circ\!\!\Rightarrow h(S)2 \circ\!\!\Rightarrow^{l+1} \beta$$

a vyjádřeme ji přesněji jako

$$S_\Gamma \circ\!\!\Rightarrow h(S)2 \circ\!\!\Rightarrow^l \alpha \circ\!\!\Rightarrow \beta,$$

kde $\alpha, \beta \in V_\Gamma^\circ$. Zdůrazněme, že α a β jsou tvaru $h(y_1)2\ldots h(y_k)2$, kde $y_j \in N$ pro $j = 1, \ldots, k$, $k \geq 0$.

Předpokládejme, že

$$\beta = h(y_1)2h(y_2)2\ldots h(y_k)2,$$

kde $y_j \in N$ pro $j = 1, \ldots, k$, $k \geq 0$. Podle indukční hypotézy

$$S \Rightarrow^{l+1} y_1y_2\ldots y_k.$$

Níže jsou rozebrány všechny možnosti jak může Γ provést derivaci

$$\alpha \circ\!\!\Rightarrow \beta.$$

Poznamenejme, že v dalších částech $u, v \in V_\Gamma^\circ$ jsou stejných tvarů jako α a β. V případě gramatiky G platí $u_G, v_G \in N^*$.

1. Nechť $2 \rightarrow \overline{h}(A) \in P_\Gamma$, kde $A \in N$. Pak

   $$\alpha = uh(X)2h(Y)2v \circ\!\!\Rightarrow uh(X)\overline{h}(A)h(Y)2v = \beta,$$

   kde $X, Y \in N$. Podřetězce $h(X)$, $h(A)$ a $h(Y)$ vyjádříme jako

   - $h(X) = X_1\ldots X_nX_n\ldots X_1$
   - $h(A) = A_1\ldots A_nA_n\ldots A_1$
   - $h(Y) = Y_1\ldots Y_nY_n\ldots Y_1$,

kde X_i, A_i, $Y_i \in N_\Gamma$ pro $i = 1, 2, \dots n$. Připomeňme, že

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

a přesněji X_i, A_i, $Y_i \in \{0, 1\}$. V bodech (1a) až (1d) rozebereme všechny možné případy týkající se X, Y a A.

(1a) Předpokládejme, že $X \neq A$ a $A \neq Y$. Nechť $X_i = A_i$ a $A_j = Y_j$ pro každé $i = 1, 2, \dots, r$ a $j = 1, 2 \dots, s$, kde $0 \leq r < n$, $0 \leq s < n$. Potom je řetězec

$$\beta = uX_1 \dots X_n X_n \dots X_1 \overline{A_1} \dots \overline{A_n A_n} \dots \overline{A_1} Y_1 \dots Y_n Y_n \dots Y_1 2v$$

redukován na

$$\beta = uX_1 \dots X_n X_n \dots X_{r+1} \overline{A_{r+1}} \dots \overline{A_n A_n} \dots \overline{A_{s+1}} Y_{s+1} \dots Y_n Y_n \dots Y_1 2v.$$

Protože redukovaná větná forma obsahuje řetězec inverzních symbolů

$$\overline{A_{r+1}} \dots \overline{A_n A_n} \dots \overline{A_{s+1}},$$

a kromě toho

$$|X_1 \dots X_n X_n \dots X_{r+1} \overline{A_{r+1}} \dots \overline{A_n A_n} \dots \overline{A_{s+1}} Y_{s+1} \dots Y_n Y_n \dots Y_1| \neq 2n$$

a P_Γ neobsahuje žádné pravidlo přepisující symboly $\overline{0}$ a $\overline{1}$, není zde žádný způsob, jak je z větné formy odstranit. Nejsme tedy schopni vygenerovat platnou větu tvořenou pouze terminálními symboly.

(1b) Nyní předpokládejme, že $X = A$ a $A \neq Y$. Řetězec β je pak redukován na

$$\beta = uY_1 \dots Y_n Y_n \dots Y_1 2v = uh(Y)2v.$$

Všimněme si, že $h(X)$ je odstraněno. Podle bodu IV konstrukce $A \to \varepsilon \in P$. Protože $A = X$, potom

$$\alpha_G = u_G XY v_G \Rightarrow u_G Y v_G = \beta_G$$

v G. Jasně také

$$S \Rightarrow^{l+1} \beta_G$$

v G a Tvrzení C platí.

(1c) Dále předpokládejme, že $X \neq A$ a $A = Y$. Potom je řetězec β redukován na

$$\beta = uX_1 \dots X_n X_n \dots X_1 2v = uh(X)2v.$$

V tomto případě je odstraněno $h(Y)$. Podle bodu IV konstrukce $A \to \varepsilon \in P$. Protože $A = Y$, pak

$$\alpha_G = u_G XY v_G \Rightarrow u_G X v_G = \beta_G$$

v G a tudíž i

$$S \Rightarrow^{l+1} \beta_G$$

v G a Tvrzení C rovněž platí.

(1d) Konečně předpokládejme, že $X = A$ a $A = Y$. Potom

$$\beta = uX_1 \dots X_n X_n \dots X_1 2v = uh(X)2v = uY_1 \dots Y_n Y_n \dots Y_1 2v = uh(Y)2v.$$

V tomto případě je odstraněno $h(X)$ nebo $h(Y)$. Podle bodu IV konstrukce $A \to \varepsilon \in P$. Jelikož $A = X = Y$, pak

$$\alpha_G = u_G XY v_G \Rightarrow u_G X v_G = u_G Y v_G = \beta_G$$

a tedy

$$S \Rightarrow^{l+1} \beta_G$$

v G. Tvrzení C i v tomto případě platí.

2. Nechť $2 \to \overline{h}(A)\overline{2}2h(B)2h(C)2 \in P_\Gamma$, kde $A, B, C \in N$. Pak

$$\alpha = uh(X)2h(Y)2v \circ\!\!\Rightarrow uh(X)\overline{h}(A)\overline{2}2h(B)2h(C)2h(Y)2v = \beta,$$

kde $X, Y \in N$. Podle Lemmatu 1 platí $A \neq B$. V bodech (2a) a (2b) dále prozkoumáme všechny možné případy týkající se X a A.

(2a) Uvažujme $X \neq A$. Všimněme si, že situace je analogická bodu (1a) a derivace platné věty je zablokována.

(2b) Nyní uvažujme, že $X = A$. V tomto případě je situace analogická bodu (1b). Výsledná větná forma je tvaru

$$\beta = uh(B)2h(C)2h(Y)2v.$$

Podle bodu III konstrukce $A \to BC \in P$, takže

$$\alpha_G = u_G XY v_G = u_G AY v_G \Rightarrow u_G BCY v_G = \beta_G.$$

Zjevně

$$S \Rightarrow^{l+1} \beta_G$$

v G a Tvrzení C platí.

3. Nechť $2 \to \overline{h}(B)\overline{2}\overline{h}(A)\overline{2}2h(C)2h(D)2 \in P_\Gamma$, kde $A, B, C, D \in N$. Potom

$$\alpha = uh(X)2h(Y)2h(Z)2v \circ\!\!\Rightarrow uh(X)2h(Y)\overline{h}(B)\overline{2}\overline{h}(A)\overline{2}2h(C)2h(D)2h(Z)2v = \beta.$$

Podle Lemmatu 1 platí $A \neq C$. V bodech (3a) až (3c) opět rozebereme všechny možné případy týkající se vzájemného vztahu symbolů X, Y, A a B.

(3a) Předpokládejme, že $Y = B$ a $X \neq A$. V tomto případě je podřetězec $2h(Y)\overline{h}(B)\overline{2}$ vymazán a výsledná (ještě ne úplně redukovaná) větná forma je tvaru

$$\beta = uh(X)\overline{h}(A)h(C)2h(D)2h(Z)2v.$$

Situace je opět analogická bodu (1a) a derivace platné věty tvořené pouze terminálními symboly je zablokována.

(3b) Nyní uvažujme, že $Y \neq B$. Výsledná ne zcela redukovaná větná forma je tvaru

$$\beta = uh(X)2h(Y)\overline{h}(B)\overline{2}\overline{h}(A)h(C)2h(D)2h(Z)2v.$$

Vyjádřeme podřetězce $h(X)$, $h(Y)$, $h(B)$, $h(A)$ a $h(C)$ jako

- $h(X) = X_1 \dots X_n X_n \dots X_1$
- $h(Y) = Y_1 \dots Y_n Y_n \dots Y_1$
- $h(B) = B_1 \dots B_n B_n \dots B_1$
- $h(A) = A_1 \dots A_n A_n \dots A_1$
- $h(C) = C_1 \dots C_n C_n \dots C_1$,

kde B_i, C_i, X_i, $Y_i \in \{0, 1\}$ pro $i = 1, 2, \dots n$. Připomeňme, že podle Lemmatu 1 je $A \neq C$. Předpokládejme, že $Y_i = B_i$ pro každé $i = 1, 2, \dots, r$ a $A_j = C_j$ pro každé $j = 1, 2, \dots, s$, kde $0 \leq r < n$ a $0 \leq s < n$. Pak je β redukována na

$$\beta = uX_1 \dots X_n X_n \dots X_1 2 Y_1 \dots Y_n Y_n \dots Y_{r+1}\overline{B_{r+1}} \dots \overline{B_n B_n} \dots$$

$$\dots \overline{B_1}\overline{2}\overline{A_1} \dots \overline{A_n A_n} \dots \overline{A_{s+1}} C_{s+1} \dots C_n C_n \dots C_1 2h(D)2h(z)2v.$$

Všimněme si, že případná rovnost či nerovnost symbolů X a A není v tomto případě důležitá. Protože redukovaná větná forma obsahuje podřetězec

$$\overline{B_{r+1}} \dots \overline{B_n B_n} \dots \overline{B_1}\overline{2}\overline{A_1} \dots \overline{A_n A_n} \dots \overline{A_{s+1}}$$

a kromě toho je konstantní délka zakódovaných nonterminálů $2n$ porušena, neexistuje způsob jak derivovat platnou větu jazyka a derivace je tudíž zablokována.

(3c) Konečně uvažujme, že $Y = B$ a $X = A$. Pak je β zredukována na

$$\beta = uh(C)2h(D)2h(Z)2v.$$

Toto je korektní větná forma. Podle bodu II konstrukce $AB \rightarrow CD \in P$, takže

$$\alpha_G = u_G XYZv_G = u_G ABZv_G \Rightarrow u_G CDZv_G = \beta_G.$$

Rozhodně

$$S \Rightarrow^{l+1} \beta_G$$

v G a Tvrzení C platí.

Tím je platnost Tvrzení C dokázána. □

Tvrzení D. Jestliže

$$h(y_1)2 \dots h(y_k)2 \circ\!\!\Rightarrow^k w$$

v Γ použitím pouze pravidel tvaru $2 \rightarrow \overline{h}(A)a$, pak

$$y_1 \dots y_k \Rightarrow^k w$$

v G, kde A, $y_i \in N$ pro $i = 1, 2, \dots k$, $k \geq 0$, $a \in T$ a $w \in T^*$. Bez ztráty obecnosti předpokládejme, že je výsledná věta generována nejlevější derivací.

Základ indukce. Nechť $k = 0$. Potom $\varepsilon \circ\!\!\Rightarrow^0 \varepsilon$ v Γ. Rozhodně také $\varepsilon \Rightarrow^0 \varepsilon$ v G.

Indukční hypotéza. Předpokládejme, že Tvrzení D platí pro všechna $k \leq l$, kde l je kladné celé číslo.

Indukční krok. Uvažujme libovolnou větnou formu tvaru

$$h(y_1)2h(y_2)2 \dots h(y_l)2h(y_{l+1})2,$$

kde $y_i \in N$ pro $i = 1, 2, \dots l + 1$. Derivaci z Tvrzení D vyjádříme jako

$$h(y_1)2h(y_2)2 \dots h(l_k)2h(y_{l+1})2 \circ\!\!\Rightarrow^k wh(y_{l+1})2 \circ\!\!\Rightarrow wa,$$

kde $w \in T^*$, $a \in T$.

Nechť $2 \rightarrow \overline{h}(y_{l+1})a \in P_\Gamma$, kde $y_{l+1} \in N$ a $a \in T$. Pak

$$wh(y_{l+1})2 \circ\!\!\Rightarrow w\underline{h(y_{l+1})\overline{h}(y_{l+1})}a = wa$$

v Γ. Podle bodu IV konstrukce $y_{l+1} \rightarrow a \in P$, takže

$$wy_{l+1} \Rightarrow wa$$

v G. Jasně také

$$y_1y_2 \dots y_ky_{l+1} \Rightarrow^{l+1} wa$$

v G.

Tvrzení D tedy platí. □

Tvrzeními C a D jsme dokázali platnost inkluze $L(\Gamma)^\circ \subseteq L(G)$.

Celkově Tvrzení A, B, C a D dokazují, že $L(G) = L(\Gamma)^\circ$, takže **CF**$^\circ$**R**=**RE** a Věta 6.2 je tímto dokázána. ■

6.3 Shrnutí

Výše uvedený důkaz demonstruje, jak lze velmi jednoduchou modifikací běžné bezkontextové gramatiky dramaticky zvýšit její vyjadřovací schopnosti až na úroveň rekurzívně vyčíslitelných jazyků. Přestože se v teorii formálních jazyků nejedná o jediný formální model založený na bezkontextových přepisovacích pravidlech, tkví jeho význam v podstatě modifikace, kterou bylo zvýšení generativní síly docíleno. Všimněme si, že zde není žádná struktura nebo omezení navíc, jak je tomu např. u gramatik z oblasti řízeného přepisování [10].

Tou jedinou a zásadní modifikací je v tomto případě prostá záměna volného monoidu za volnou grupu, nad kterou je definována relace přímé derivace. Kromě toho jsme vhodným kódováním nonterminálních symbolů dosáhli i jejich výrazné redukce.

Kapitola 7

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 8

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.

8.1 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.

Zamysleme se na samém závěru této části nad možnostmi dalšího výzkumu těchto automatů. V našem pojednání jsme se zabývali pouze jejich nedeterministickými variantami. Hned se tedy nabízí studium deterministických oboustranných zásobníkových automatů. Otázkou zůstává, zda bude jejich síla zachována (podobně jako je tomu v případě Turingových strojů), nebo zda bude snížena. My jsme v této práci studovali pouze vyjadřovací schopnosti. Dalším zajímavým a rozhodně přínosným tématem by mohlo být studium paměťové a časové náročnosti těchto automatů.

8.2 Vertikální kontext v obecných gramatikách

Jak jsme se již zmínili dříve, v běžných 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.

Důkaz týkající se vyjadřovacích schopností takto omezených gramatik vede na konstrukci konečného automatu. Stavy tohoto automatu jsou vícesložkové. Každá z těchto složek hraje důležitou roli při simulaci jednotlivých derivačních kroků z původní gramatiky. V konečném důsledku jsme tedy degradovali schopnosti gramatik v Kurodově normální formě až na úroveň regulárních jazyků. To je velmi překvapivé zjištění, neboť jsme se ze samého vrcholu Chomského hierarchie dostali až na její dno.

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 $i = 2, 3, ...$. 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 [15] a [29]. 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.

8.3 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 [19] nebo [12]. 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.

Zamysleme se na samý závěr nad možnými dalšími směry výzkumu těchto gramatik. Jak jsme se již zmínili v předchozích odstavcích, i zde se na prvním místě nabízí studium časové a paměťové složitosti těchto gramatik. Zde by mohl být jedním z hledisek například počet nonterminálních symbolů výchozí gramatiky v Kurodově normální formě, případně i délka generovaného řetězce. Vzhledem k tomu, že tyto gramatiky pracují nedeterministicky (v daném kroku nelze jednoznačně určit, který symbol ve větné formě přepsat, aby nebyla derivace zablokována), bylo by rozhodně přínosné snažit se tento proces určitým způsobem omezit. Protože jsou všechna přepisovací pravidla bezkontextová, bylo by možné inspirovat se vlastnostmi běžných bezkontextových gramatik a definovat pojem nejlevější resp. nejpravější derivace, kdy by byl v každém kroku přepsán vždy nejlevější resp. nejpravější nonterminální symbol (v našem případě symbol 2). To však ještě nezaručí to, že by bylo dané přepisovací pravidlo aplikováno správným způsobem tak, aby došlo ke korektní redukci vzniklého inverzního podřetězce. Součástí výzkumu by tedy mohlo být i nalezení algoritmu pro určení pozic nonterminálních symbolů, které mohou být korektně přepsány daným přepisovacím pravidlem. Z tohoto zjištění by pak byl přepsán podle typu derivace vždy symbol na nejlevější, resp. nejpravější pozici.

Uvedené myšlenky však dalece překračují rozsah této práce a tvoří tedy pouze možné náměty pro budoucí výzkum.

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 and P. Blatný. How to generate recursively enumerable languages using only context-free productions and eight nonterminals. In *Proceedings of 11th Conference and Competition Student EEICT 2005, Volume 3*, pages 536–541. Faculty of Electrical Engineering and Communication BUT, 2005.

[4] R. Bidlo, P. Blatný, and A. Meduna. Automata with two-sided pushdowns defined over free groups generated by reduced alphabets. *Kybernetika*, 2007. in press.

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

[6] R. L. Bisbey. Finite state queue automata, 1969.

[7] P. Blatný. E0l grammars on free groups. In *Proceedings of 8th International Conference ISIM'05 Information System Implementation and Modeling*, pages 81–86. MARQ, 2005.

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

[9] J. Dassow and V. Mitrana. Finite automata over free groups. *Int. Journal of Algebra and Computation*, 10(6):725–738, 2000.

[10] J. Dassow and G. Paun. *Regulated Rewriting in Formal Language Theory.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1990.

[11] A. Drápal. *Teorie grup - základní aspekty.* Karolinum, Praha, 2000.

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

[13] S. A. Greibach. A new normal-form theorem for context-free phrase structure grammars. *J. ACM*, 12(1):42–52, 1965.

[14] M.A. A. Harrison and M. A. Harrison. *Introduction to Formal Language Theory.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1978.

[15] 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.

[16] N. Jacobson. *Basic algebra.* W. H. Freeman and Company, New York, second edition, 1989.

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

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

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

[20] S. Y. Kuroda. Classes of languages and linear-bounded automata. *Information and Control*, 7(2):207–223, 1964.

[21] A. G. Kuroš. *Kapitoly z obecné algebry.* Academia, Praha, 1977.

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

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

[24] A. Meduna and M. Švec. *Grammars with Context Conditions and Their Applications [Wiley, 2004].* John Wiley & Sons, 2005.

[25] J. Milne. Group theory, 2003. cit. 2007-03-24.

[26] G. Paun. Grammar systems: A grammatical approach to distribution and cooperation. In *ICALP '95: Proceedings of the 22nd International Colloquium on Automata, Languages and Programming*, pages 429–443, London, UK, 1995. Springer-Verlag.

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

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

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

[30] M. Sipser. *Introduction to the Theory of Computation.* International Thomson Publishing, 1996.

Seznam symbolů a zkratek

$\Box$	konec části důkazu
$\blacksquare$	konec důkazu
$\emptyset$	prázdná množina
$\cup$	sjednocení množin
$\cap$	průnik množin
$\times$	kartézský součin
$\in$	náležející; je prvkem množiny
$\notin$	nenáležející; není prvkem množiny
$\subset$	je vlastní podmnožinou
$\subseteq$	je podmnožinou
$\Rightarrow$	symbol relace přímé derivace; derivuje
$\Rightarrow^*$	reflexivní a tranzitivní uzávěr relace $\Rightarrow$; derivuje v nula nebo více krocích
$\Rightarrow^+$	tranzitivní uzávěr relace $\Rightarrow$; derivuje v jednom nebo více krocích
$\Rightarrow^n$	$n-$tá mocnina relace $\Rightarrow$, $n \geq 0$; derivuje v n krocích
$\circ\!\!\Rightarrow$	symbol relace přímé derivace nad volnou grupou; derivuje nad volnou grupou
$\circ\!\!\Rightarrow^*$	reflexivní a tranzitivní uzávěr relace $\circ\!\!\Rightarrow$; derivuje v nula nebo více krocích nad volnou grupou
$\circ\!\!\Rightarrow^+$	tranzitivní uzávěr relace $\circ\!\!\Rightarrow$; derivuje v jednom nebo více krocích nad volnou grupou
$\circ\!\!\Rightarrow^n$	$n-$tá mocnina relace $\circ\!\!\Rightarrow$, $n \geq 0$; derivuje v n krocích nad volnou grupou
$\vdash$	symbol relace přechodu; přechází do
$\vdash^*$	reflexivní a tranzitivní uzávěr relace $\vdash$; přechází v nula nebo více krocích do
$\vdash^+$	tranzitivní uzávěr relace $\vdash$; přechází v jednom nebo více krocích do
$\vdash^n$	$n-$tá mocnina relace $\vdash$, $n \geq 0$; přechází v n krocích do
$\rightarrow$	symbol přepsání/přechodu; přepiš na/přejdi do
$=$	symbol rovnosti; je rovno
$\neq$	symbol nerovnosti; není rovno; různé od
$\leq$	menší nebo rovno
$\geq$	větší nebo rovno
$\cdot$	spojení, konkatenace
$-$	operace odčítání; záporné znaménko; operátor opačné hodnoty
$\overline{a}$	nad volnou grupou značí inverzní symbol k symbolu a
$\lceil n \rceil$	n zaokrouhleno na nejbližší vyšší číslo
$\vert$	oddělovač

$\|w\|$	délka řetězce w
α	řecké písmeno *alfa*
β	řecké písmeno *beta*
Γ, γ	řecké písmeno *gama*
ε	řecké písmeno *epsilon*; symbol prázdného řetězce
λ	řecké písmeno *lambda*
μ	řecké písmeno *mý*
ρ	řecké písmeno *ró*
Σ	řecké písmeno *sigma*
Φ	řecké písmeno *fí*
ω	řecké písmeno *omega*
$^\circ$	symbol volné grupy, není-li v kontextu uvedeno jinak
*	symbol volného monoidu, není-li v kontextu uvedeno jinak; iterace
$^+$	pozitivní iterace, není-li v kontextu uvedeno jinak
$\sim$	symbol ekvivalence; je ekvivalentní s
$\hookleftarrow$	posun čtecí hlavy vlevo
$\hookrightarrow$	posun čtecí hlavy vpravo
$\downarrow$	zachování pozice čtecí hlavy
$\triangle$	prázdný symbol
2sPDA	třída jazyků přijímaných oboustrannými zásobníkovými automaty
2sPDAR	třída jazyků přijímaných oboustrannými zásobníkovými automaty s redukovaným počtem symbolů zásobníkové abecedy
CF	třída bezkontextových jazyků
CF$^\circ$R	třída jazyků generovaných bezkontextovými gramatikami nad volnými grupami s redukovaným počtem nonterminálů; zkratka pro označení těchto gramatik
CS	třída kontextových jazyků
FA	třída jazyků přijímaných konečnými automaty
LBA	třída jazyků přijímaných lineárně ohraničenými automaty
LEQG	třída jazyků generovaných levě rozšířenými frontovými gramatikami
PDA	třída jazyků přijímaných zásobníkovými automaty
PDA*(Empty)*	třída jazyků přijímaných zásobníkovými automaty s vyprázdněním zásobníku
PDA*(Final)*	třída jazyků přijímaných zásobníkovými automaty přechodem do koncového stavu
PDA*(Empty&Final)*	třída jazyků přijímaných zásobníkovými automaty přechodem do koncového stavu a s vyprázdněním zásobníku
QG	třída jazyků generovaných frontovými gramatikami
RE	třída rekurzívně vyčíslitelných jazyků
REG	třída regulárních jazyků
TM	třída jazyků přijímaných Turingovými stroji