

# DETERMINISTIC LL MATRIX GRAMMAR OF TYPE 2

**Petr Bednář**

Bachelor Degree Programme (3), FIT BUT

E-mail: xbedna46@stud.fit.vutbr.cz

Supervised by: Alexander Meduna

E-mail: meduna@fit.vutbr.cz

**Abstract:** This article discusses a version of restricted matrix grammar that can be accepted deterministically using a type of left most derivation. I introduce algorithm, that accepts language defined by this type of grammar. Then I discuss generative strength of this type of grammar.

**Keywords:** Matrix grammar, deterministic, regulated rewriting

## 1 ÚVOD

V této práci se věnuji omezenému typu maticových gramatik, která umožňuje omezeným způsobem skákat mezi neterminály a je deterministická. Tato práce předpokládá základní znalosti z oboru formálních jazyků.

## 2 VÝCHOZÍ DEFINICE

V práci dále používám gramatického automatu pro vyjádření řízených derivací maticových gramatik.

**Definice 1:** Gramatický automat je konečný automat, který obsahuje jeden počáteční stav, který neaplikuje žádná gramatická pravidla a další stavy pro aplikaci pravidel gramatiky, kterou vyjadřuje. Přechody tohoto automatu jsou značeny dvojicí. Řetězcem prepisovaným, na který bude výsledné pravidlo aplikováno, a podmínkou která dovoluje použití pravidla. Posloupnost derivací je úspěšná pouze za podmínky, že automat skončí v koncovém stavu.

Maticová gramatika je bezkontextová gramatika rozšířená o sekvence pravidel, matice, kde prvky matice musejí být aplikovány všechny a těsně za sebou v předepsaném pořadí.

**Definice 2:** Maticová gramatika [1] je dvojice

$$H = (G, M),$$

kde,  $G = (N, T, P, S)$  je bezkontextová gramatika,  $N$  je abeceda neterminálů,  $S \in N$  je počáteční neterminál,  $T$  je abeceda terminálů,  $P$  je konečná množina pravidel tvaru  $A \rightarrow x$ , kde  $A \in N$ ,  $x \in (N \cup T)^*$ , a  $M \subseteq P^+$  je konečný jazyk, jehož řetězce se nazývají matice.

Maticovou gramatiku, vyjádříme jako gramatický automat tak, že každá matice pravidel je převedena na řetězec stavů automatu. Počátečním stavem řetězce je počáteční stav gramatického automatu. Další stavy řetězce poté odpovídají jednotlivým prvkům matice. Stav na konci tohoto řetězce je koncovým stavem. Tento řetězec je obohacen o prázdnou hranu z koncového stavu do stavu počátečního.

Práce se zabývá úpravou maticové gramatiky typu 2 [2] využívající nejlevější derivace pomocí nejlevějších terminálů.

**Definice 3:** Maticová gramatika typu 2 využívající nejlevější derivace a nejlevější terminály, dále jen  $MAT_{LL2}^\lambda$ , je maticová gramatika, kde je v každém derivačním kroku přepsán nejlevější neterminál, který je možno přepsat na základě právě uplatnitelných pravidel.

### 3 ODSTRANĚNÍ NEDETERMINISMU

Při hledání derivačního stromu věty jazyka definovaného maticovou gramatikou narážíme na několik zdrojů nedeterminismu. Deterministická gramatika musí zaručit, že pro každou větnou formu je uplatnitelné nejvýše jedno pravidlo a toto pravidlo je uplatnitelné na jediný neterminál. Typ 2 sám pokrývá podmínku uplatnění pravidla na jediný možný neterminál. Podmínku možnosti použít nejvýše jedno pravidlo musíme splnit další úpravou gramatiky.

Maticová gramatika nám dovoluje využít jakékoli pravidlo, které je dosažitelné ze současného stavu gramatického automatu a odpovídá přepisovanému neterminálu. Mezi těmito pravidly se dále rozhodujeme na základě terminálů, přítomných ve vstupním řetězci. Pro výběr pravidla gramatiky, využívající nejlevější derivace, je využít nejlevější terminál, který dovoluje použít libovolné pravidlo z uvažovaných.

Indikace možnosti použít pravidlo, které samo neobsahuje potřebný terminál, je u deterministických bezkontextových gramatik odvozena od rekurzivního prohledávání dalších pravidel. Skoky, kdy je možno přepisovat neterminály v různém pořadí, však tuto možnost vylučují, jelikož není zaručeno pořadí stavů gramatického automatu pro produkci jediného neterminálu. Proto musíme vyloučit možnost takového prohledávání úpravou samotných pravidel.

**Definice 4:** Gramatika  $MAT_{LL2S}^\lambda$  je taková  $MAT_{LL2}^\lambda$  gramatika, kde pravá strana každého pravidla je ve formátu  $tn$ , kde  $t \in T$  a  $n \in N^*$ , nebo se jedná o  $\varepsilon$ -pravidlo, a lze sestavit deterministický gramatický automat, kde je přepisovaným řetězcem levá strana pravidla a podmínkou provedení hrany je terminál přímo produkovaný pomocí pravidla, jinak je podmínka prázdná.

Tato gramatika přepisuje neterminál dle nejlevějšího terminálu, který je produkován z daného neterminálu.

**Definice 5:** Gramatika  $MAT_{LL2R}^\lambda$  je taková  $MAT_{LL2}^\lambda$  gramatika, kde pravá strana každého pravidla je ve formátu  $ntm$ , kde  $t \in T$  a  $n, m \in N^*$ , nebo se jedná o  $\varepsilon$ -pravidlo, a lze sestavit deterministický gramatický automat, kde je přepisovaným řetězcem levá strana pravidla a podmínkou provedení hrany je terminál přímo produkovaný pomocí pravidla, jinak je podmínka prázdná.

Tato gramatika přepisuje neterminál dle nejlevějšího terminálu obsaženého v přepisovaném řetězci, který je možno přepsat současnými pravidly

### 4 GENERATIVNÍ SÍLA

Deterministická  $MAT_{LL2S}^\lambda$  umožňuje vytvořit gramatiku pro jazyk  $L_1 = \{a^n b^n c^n | n \geq 0\}$ , který nenáleží do třídy bezkontextových jazyků. Avšak nejsme schopni vytvořit takovou deterministickou  $MAT_{LL2S}^\lambda$  gramatiku, umožňující rozpoznat jazyk  $L_2 = \{ww^r | w \in \{a, b\}^*\}$ .

**Teorem 1:**  $MAT_{LL2S}^\lambda \subset MAT_{LL2R}^\lambda$

*Důkaz.* Každá gramatika  $MAT_{LL2S}^\lambda$  je zároveň gramatikou  $MAT_{LL2R}^\lambda$ . Mějme jazyk  $L$  generovaný  $MAT_{LL2R}^\lambda$  gramatikou  $H = (G, M)$ , kde

$$G = (\{S, A, B\}, \{a, b\}, R, S)$$

$$R = \{S \rightarrow aAB, S \rightarrow bAB, A \rightarrow aA, A \rightarrow bA, A \rightarrow \varepsilon, B \rightarrow Ba, B \rightarrow Bb, B \rightarrow \varepsilon\}$$

$$M = \{r_1 r_6, r_2 r_7, r_3 r_6, r_4 r_7, r_5 r_8\}$$

Jazyk  $L$  je jazyk  $\{ww^r | w \in \{a, b\}^+\}$ , který není ve třídě jazyků rozpoznatelných pomocí  $MAT_{LL2S}^\lambda$  gramatik. ■

## 5 ALGORITMUS

Větná forma je rozdělena na jednotlivé sekce a počáteční větná forma má jedinou sekci obsahující startovací symbol. Každé uplatnění pravidla rozdělí sekci, ve které je nahrazovaný neterminál na tři části. Část před terminálem pravidla, terminál a část za terminálem pravidla. Pro každý symbol ve vstupním řetězci je zaznamenána příslušnost k sekci. Sekce zajišťují lokalitu uplatňování pravidel vzhledem ke vstupním symbolům.

Pro vybraný neterminál je v příslušné skupině nalezen nejlevější terminál vstupního řetězce, který indikuje použití vybraného pravidla. Pokud přepisujeme první neterminál skupiny a pravá strana pravidla má jako první symbol terminál je prohledávání omezeno na první vstupní terminál skupiny. Naopak, pokud přepisujeme poslední neterminál skupiny a pravá strana pravidla má terminál jako poslední symbol je prohledávání omezeno na poslední vstupní terminál skupiny. Pokud takový terminál neexistuje a existuje  $\epsilon$ -pravidlo pro vybraný neterminál, je toto pravidlo uplatněno.

Tento algoritmus provede jeden krok derivace pomocí nejlevějších gramatik typu 2. Krok se provádí do té doby, než nastane chyba nebo je současná větná forma prázdná.

### Algoritmus 1:

- 1: Vyber aktuální množinu uplatnitelných pravidel z automatu
- 2: Najdi nejlevější neterminál v současné větě, který je možno přepsat vybranými pravidly
- 3: **if** Takový neterminál neexistuje **then**
- 4:     Chyba
- 5: **else**
- 6:     Vyber pouze pravidla přepisující nalezený neterminál
- 7:     Hledej první terminál umožňující přechod automatu vybranými pravidly
- 8:     **if** Terminál nenalezen **then**
- 9:         **if** Možno smazat neterminál **then**
- 10:             Smaž neterminál
- 11:             goto 17
- 12:         **else**
- 13:             Chyba
- 14:         **end if**
- 15:     **end if**
- 16:     Přepiš neterminál a rozděl sekci
- 17:     Proved' krok gramatického automatu
- 18:     Eliminuj terminály v současné větné formě porovnáním se vstupem
- 19: **end if**

## 6 ZÁVĚR

Prozkoumal jsem možnost jednoznačného přijímání jazyků generovaných maticovými gramatikami, které nejsou omezeny na striktně nejlevější neterminál a našel jsem takovou třídu gramatik, která umožňuje generování jazyků, které spadají do jazyků kontextových, a k ní korespondující deterministický algoritmus přijímání řetězce.

## REFERENCE

- [1] Meduna, A. and Zemek, P.: Regulated Grammars and Their Transformations, Brno University of Technology 2010, s. 29-30, 978-80-214-4203-0
- [2] Dassow, J. and Păun, G.: Regulated Rewriting in Formal Language Theory, Springer-Verlag New York, Inc. 1989, s. 54, ISBN 0-387-51414-7