Vysoké učení technické v Brně

Fakulta informačních technologií

Gramatiky nad volnými grupami

2005 Petr Blatný

Abstrakt

Tento dokument zavádí pojmy bezkontextové gramatiky nad volnou grupou a E0L gramatiky nad volnou grupou. Dále zkoumá schopnosti nově vzniklých struktur generovat jazyky. Úvodem jsou připomenuty nezbytné pojmy a definice, jak z oblasti formálních jazyků, tak i z oblasti algebraických struktur. Jsou definovány nové struktury bezkontextové a E0L gramatiky, ke kterým je přidána volná grupa generovaná totální abecedou těchto gramatik. Kombinací gramatiky a volné grupy získáme strukturu jejíž generativní síla odpovídá třídě jazyků typu 0.

Klíčová slova

bezkontextová gramatika, E0L gramatika, derivace, volná grupa, bezkontextová gramatika nad volnou grupou, E0L gramatika nad volnou grupou, Kurodova normální forma, Chomského hierarchie

Obsah

1 Úvod **4**

2 Základy teorie formálních jazyků **5**
2.1 Abeceda 5
2.2 Řetězec nad abecedou 5
2.3 Gramatika 5
2.4 Chomského klasifikace gramatik 7
2.4.1 Typ 0 7
2.4.2 Typ 1 7
2.4.3 Typ 2 8
2.4.4 Typ 3 8
2.5 Kurodova a Pentonnenova normální forma 8

3 L systémy **10**
3.1 0L systém 10
3.2 D0L systém 10
3.3 P0L systém 10
3.4 E0L systém 11

4 Úvod do základních algebraických struktur **13**
4.1 Finitární algebraické operace 13
4.2 Axiomy binárních operací 13
4.3 Grupoidy, pologrupy, monoidy a grupy 14

5 Volné grupy **15**
5.1 Volný monoid 15
5.2 Volná grupa 15

6 Gramatiky nad volnými grupami **16**
6.1 Bezkontextové gramatiky nad volnými grupami 16
6.2 Derivace v bezkontextové gramatice nad volnou grupou 16
6.3 Jazyk generovaný bezkontextovou gramatikou nad volnou grupou 17
6.4 Generativní schopnosti bezkontextových gramatik nad volnými grupami . . . 17
6.5 E0L gramatiky nad volnými grupami 19
6.6 Derivace v E0L gramatice nad volnou grupou 20
6.7 Jazyk generovaný E0L gramatikou nad volnou grupou 20
6.8 Generativní schopnosti E0L gramatik nad volnými grupami 20

7 Závěr **22**

1 Úvod

Teorie formálních jazyků představuje jednu z důležitých oblastí informatiky. Základy této vědní disciplíny založil americký matematik Noam Chomsky, který je autorem definice gramatiky formálního jazyka a klasifikace formálních jazyků (Chomského hierarchie jazyků).

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

Na samém vrcholu Chomského hierarchie jsou tzv. jazyky tupu 0. V některé literatuře se můžeme též setkat s označením rekurzivně vyčíslitelné jazyky, frázově strukturované jazyky nebo též jazyky přijímané Turingovými stroji. Všechny výše uvedená označení jsou si ekvivalentní.

Standardně jsou derivace v gramatikách definovány nad volným monoidem generovaným množinami terminálních a nonterminálnich symbolů (totální abecedou) operací konkatenace. Avšak v některých studií jsou tyto derivace definovány nad jinými algebraickými strukturami (viz. [5], [2], [1]). V této práci provedeme modifikaci derivace a to tak, že ji budeme definovat nad volnou grupou. Dále se budeme zabývat generativní silou takto modifikovaných gramatik.

Tento dokument pojednává o návrhu struktur, které jsou schopny generovat celou třídu rekurzivně vyčíslitelných jazyků. Jako základ použijeme bezkontextovou gramatiku a E0L gramatiku. Společně definujeme volnou grupu nad totální abecedou těchto gramatik. Využitím schopností uvedených prostředků dokážeme, že bezkontextové gramatiky nad volnými grupami a E0L gramatiky nad volnými grupami definují třídu jazyků typu 0.

2 Základy teorie formálních jazyků

Základní pojmy pro vymezení jazyka jsou *abeceda* a *řetězec.*

2.1 Abeceda

Definice 2.1 Abecedou rozumíme neprázdnou množinu prvků. Tyto prvky nazýváme *symboly abecedy.*

2.2 Řetězec nad abecedou

Definice 2.2 *Řetězcem* (nebo také *větou* či *slovem*) nad danou abecedou rozumíme každou konečnou posloupnost symbolů abecedy. Prázdnou posloupnost symbolů, tj. posloupnost, která neobsahuje žádný symbol, nazýváme *prázdný řetězec.* Prázdný řetězec budeme zapisovat symbolem ε.

Definice 2.3 Nechť x, y jsou řetězce nad abecedou Σ. *Konkatenací* (*zřetězením*) řetězce x s řetězcem y vznikne řetězec xy. Operace konkatenace je asociativní, tj. $x(yz) = (xy)z$, není však komutativní, $xy \neq yx$ pro $x \neq y$.

Definice 2.4 Nechť w je řetězec nad abecedou Σ. Řetězec z se nazývá *podřetězcem* řetězce w, jestliže existují řetězce x a y takové, že $w = xzy$. Řetězec x_1 se nazývá *prefixem* (*předponou*) řetězce w, jestliže existuje řetězec y_1 takový, že $w = x_1y_1$. Analogicky, řetězec y_2 se nazývá *sufixem* (*příponou*) řetězce w, jestliže existuje řetězec x_2 takový, že $w = x_2y_2$. Je-li $y_1 \neq \varepsilon$, resp. $x_2 \neq \varepsilon$, pak x_1 je *vlastní prefix*, resp. x_2 je *vlastní sufix* řetězce w.

Definice 2.5 *Délka řetězce* je nezáporné celé číslo, které udává počet symbolů řetězce. Délku řetězce x značíme $|x|$. Je-li $x = a_1a_2 \ldots a_n, a_i \in \Sigma, 1 \leq i \leq n$, pak $|x| = n$. Délka prázdného řetězce je nulová, tj. $|\varepsilon| = 0$.

Konvence 2.1 Řetězec nebo podřetězec, stávající se právě z k výskytů symbolu a, budeme symbolicky značit a^k. Například:

$$a^3 = aaa, \quad b^1 = b, \quad c^0 = \varepsilon$$

Definice 2.6 Nechť Σ je abeceda. Označme symbolem Σ^* množinu všech řetězců nad abecedou Σ včetně řetězce prázdného, symbolem Σ^+ všechny řetězce vyjma řetězce prázdného, tj. $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$. Množinu L, pro níž platí $L \subseteq \Sigma^*$ (případně $L \subseteq \Sigma^+$ pokud $\varepsilon \notin L$), nazýváme *jazykem* L nad abecedou Σ. jazykem tedy může být libovolná podmnožina řetězců nad danou abecedou.

2.3 Gramatika

Nejznámějším prostředkem pro popis jazyků je *gramatika.* Gramatika splňuje základní požadavek na reprezentaci konečných i nekonečných jazyků a požadavek konečnosti reprezentace.

Gramatika používá dvou disjunktních abeced:

(1) množiny N *nonterminálních symbolů*

(2) množiny Σ *terminálních symbolů*

Nonterminální symboly, krátce *nonterminály*, mají roli pomocných proměnných, které označují určité syntaktické celky (kategorie).

Množina *terminálních symbolů*, krátce *terminálů*, je identická s abecedou, nad níž je definován jazyk. Sjednocení obou množin, tj. $N \cup \Sigma$, nazýváme *slovníkem gramatiky* nebo také *totální abecedou gramatiky.*

Konvence 2.2 Pro zápis terminálních a nonterminálních symbolů a řetězců tvořených těmito symboly budeme používat této konvence:

(1) a, b, c, d, e reprezentují terminální symboly

(2) A, B, C, D, S, X, Y reprezentují nonterminální symboly, X a Y budou představovat pomocné nonterminály

(3) U, V, Z reprezentují terminální nebo nonterminální symboly

(4) $\alpha, \beta, \ldots, \omega, x, y, z$ reprezentují řetězce terminálních a nonterminálních symbolů

(5) u, v reprezentují řetězce pouze terminálních symbolů

Gramatika představuje generativní systém, ve kterém lze z jistého vyznačeného nonterminálu generovat, aplikací tzv. přepisovacích pravidel, řetězce tvořené pouze terminálními symboly. Takové řetězce reprezentují právě věty gramatikou definovaného jazyka.

Jádrem gramatiky je tak konečná množina *přepisovacích pravidel.* Každé přepisovací pravidlo má tvar uspořádané dvojice řetězců (α, β), stanovuje možnou substituci řetězce β namísto řetězce α, který se vyskytuje jako podřetězec generovaného řetězce. Řetězec α obsahuje alespoň jeden nonterminální symbol, řetězec β je prvkem množiny $(N \cup \Sigma)^*$. Formálně vyjádřeno, množina přepisovacích pravidel je podmnožinou kartézského součinu:

$$(N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$$

Definice 2.7 Gramatika G je čtveřice $G = (N, \Sigma, P, S)$, kde

- N je konečná množina nonterminálních symbolů
- Σ je konečná množina terminálních symbolů, $N \cap \Sigma = \emptyset$
- P je konečná podmnožina kartézského součinu $(N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$
- $S \in N$ je počáteční (také startovací) symbol gramatiky

V tomto textu se také setkáme s definicí gramatiky $G = (V, \Sigma, P, S)$, kde V je totální abecedou a platí, že množina nonterminálních symbolů je rovna $V - \Sigma$.

Prvek (α, β) množiny P nazýváme *přepisovacím pravidlem* (krátce *pravidlem*) a budeme jej zapisovat ve tvaru $\alpha \to \beta$. Řetězec α resp. β nazýváme *levou* resp. *pravou stranou* přepisovacího pravidla.

Definice 2.8 Nechť $G = (N, \Sigma, P, S)$ je gramatika a nechť λ a μ jsou řetězce z $(N \cup \Sigma)^*$. Mezi řetězci λ a μ platí relace $\Rightarrow_G$, nazývaná *přímá derivace*, jestliže můžeme řetězce λ a μ vyjádřit ve tvaru

$$\lambda = \gamma\alpha\delta$$
$$\mu = \gamma\beta\delta$$

γ a δ jsou libovolné řetězce z $(N \cup \Sigma)^*$ a $\alpha \to \beta$ je přepisovací pravidlo z P.

Platí-li mezi řetězci λ a mu relace přímé derivace, pak píšeme $\lambda \Rightarrow_G \mu$ a říkáme, že řetězec μ lze přímo generovat z řetězce λ v gramatice G. Je-li z kontextu zřejmé, že jde o derivaci v gramatice G, pak nemusíme specifikaci gramatiky pod symbolem $\Rightarrow$ uvádět.

Definice 2.9 Nechť $G = (N, \Sigma, P, S)$ je gramatika a λ a μ jsou řetězce z $(N \cup \Sigma)^*$. Mezi řetězci λ a μ platí relace $\Rightarrow^+$ nazývaná *derivace*, jestliže existuje posloupnost přímých derivací $v_{i-1} \Rightarrow v_i$, $1 \leq i \leq n$ taková, že platí:

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

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

Definice 2.10 Jestliže v gramatice G platí pro řetězce λ a μ relace $\lambda \Rightarrow^+ \mu$ nebo identita $\lambda = \mu$, pak píšeme $\lambda \Rightarrow^* \mu$. Relace $\Rightarrow^*$ je tranzitivním a reflexivním uzávěrem relace přímé derivace $\Rightarrow$.

Definice 2.11 Nechť $G = (N, \Sigma, P, S)$ je gramatika. Řetězec $\alpha \in (N \cup \Sigma)^*$ nazýváme *větnou formou*, jestliže platí $S \Rightarrow^* \alpha$, tj. řetězec α je generovatelný ze startovacího symbolu S. Větná forma, která obsahuje pouze terminální symboly, se nazývá *věta*. *Jazyk* $L(G)$, generovaný gramatikou G, je definován množinou všech vět

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

2.4 Chomského klasifikace gramatik

Chomského klasifikace gramatik (a jazyků), známá také pod názvem Chomského hierarchie jazyků, vymezuje čtyři typy gramatik podle tvaru přepisovacích pravidel, jež obsahuje množina přepisovacích pravidel. Tyto typy se označují jako typ 0, typ 1, typ 2 a typ 3.

2.4.1 Typ 0

Gramatika typu 0 obsahuje pravidla v nejobecnějším tvaru:

$$\alpha \to \beta; \quad \alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*, \quad \beta \in (N \cup \Sigma)^*$$

Tyto gramatiky se také nazývají *gramatikami neomezenými*. Jazyky definované těmito gramatikami se nazývají rekurzivně vyčíslitelné a často se označují zkratkou **RE** (z angl. recursively enumerable).

2.4.2 Typ 1

Gramatika typu 1 obsahuje pravidla tvaru:

$$\alpha A \beta \to \alpha \gamma \beta; \quad A \in N \quad \alpha\beta \in (N \cup \Sigma)^*, \quad \gamma \in (N \cup \Sigma)^+$$

nebo

$$S \to \varepsilon$$

Gramatiky typu 1 se nazývají také *gramatikami kontextovými* a to z toho důvodu, že tvar pravidla této gramatiky implikuje, že nontermunál A může být přepsán řetězcem γ pouze tehdy, je-li jeho pravým kontextem řetězec β a levým kontextem řetězec α.

Jazyk generovaný gramatikou typu 1 se nazývá jazykem kontextovým a často se označuje zkratkou **CS** (z angl. context-sensitive).

2.4.3 Typ 2

Gramatika typu 2 obsahuje pravidla tvaru:

$$A \to \gamma; \quad A \in N, \quad \gamma \in (N \cup \Sigma)^*$$

Gramatiky typu 2 se nazývají *bezkontextovými gramatikami*, protože substituci levé strany pravidla (γ) za nonterminál A lze provádět bez ohledu na kontext, ve kterém je nonterminál A uložen. Na rozdíl od kontextových gramatik, bezkontextové gramatiky smí obsahovat pravidla tvaru $A \to \varepsilon$.

Jazyk generovaný gramatikou typu 2 se nazývá jazykem bezkontextovým a často se označuje zkratkou **CF** (z angl. context-free).

2.4.4 Typ 3

Gramatika typu 3 obsahuje pravidla tvaru:

$$A \to \gamma; \quad A \in N, \quad \gamma \in \Sigma^* \cup \Sigma^* N \Sigma^*$$

Gramatika s tímto tvarem pravidel se nazývá *lineární gramatika*. K takovéto gramatice lze sestrojit ekvivalentní speciální *pravou lineární gramatiku*:

$$A \to \gamma; \quad A \in N, \quad \gamma \in \Sigma^* \cup \Sigma^* N$$

nebo *levou lineární gramatiku*:

$$A \to \gamma; \quad A \in N, \quad \gamma \in \Sigma^* \cup N \Sigma^*$$

Názvy pravé a levé lineární gramatiky jsou odvozeny od pozice nonterminálu na pravé straně pravidel. Pro tyto gramatiky je dále možné sestrojit ekvivalentní gramatiky, které se nazývají *regulární*. Opět podle pozice nonterminálu na pravé straně pravidel, rozlišujeme *pravou regulární gramatiku*:

$$A \to \gamma, \quad A \in N, \quad \gamma \in \Sigma \cup \Sigma N \cup \{\varepsilon\}$$

a *levou regulární gramatiku*:

$$A \to \gamma, \quad A \in N, \quad \gamma \in \Sigma \cup N \Sigma \cup \{\varepsilon\}$$

Gramatiky typu 3 se proto také nazývají regulárními gramatikami.

2.5 Kurodova a Pentonnenova normální forma

Pro popis jazyků typu 0 lze použít některou z normálních forem. V tomto dokumentu pro nás bude důležitá Kurodova normální forma a Pentonnenova normální forma.

Definice 2.12 Gramatika $G = (N, \Sigma, P, S)$ je v *Kurodově normální formě*, pokud její množina přepisovacích pravidel P obsahuje pouze pravidla tvaru:

- $AB \to CD$,
- $A \to BC$,
- $A \to a$,
- $A \to \varepsilon$

$A, B, C, D \in N$, $a \in T$, ε je prázdný řetězec.

Definice 2.13 Gramatika $G = (N, \Sigma, P, S)$ je v *Pentonnenově normální formě*, pokud její množina přepisovacích pravidel P obsahuje pouze pravidla tvaru:

- $AB \rightarrow AC$,
- $A \rightarrow BC$,
- $A \rightarrow a$,
- $A \rightarrow \varepsilon$

$A, B, C \in N$, $a \in T$, ε je prázdný řetězec.

Věta 2.1 Každou gramatiku G typu 0 je možné transformovat na ekvivalentní gramatiku H v Kurodově normální formě takovou, že $L(G) = L(H)$.

Věta 2.2 Každou gramatiku G typu 0 je možné transformovat na ekvivalentní gramatiku H v Pentonnenově normální formě takovou, že $L(G) = L(H)$.

3 L systémy

L systémy poprvé představil Aristid Lindenmayer v roce 1968 [8]. Původně se jednalo o paralelní přepisovací systém modelující vícebuněčné organismy. Základní myšlenka těchto systémů našla využití i v teoretické informatice. Vznikají nové jazyky založené na 0L, D0L, P0L, E0L a dalších gramatikách. Zvláštností většiny tříd jazyků, které jsou přijímány gramatikami založenými na 0L systémech, je jejich pozice v Chomského hierarchii - jednotlivé třídy mohou překrývat a přitom žádnou z nich nepokrývají.

Nás bude zajímat třída jazyků přijímaná E0L gramatikami, která leží nad třídou bezkontextových jazyků, ale nepokrývá celou třídu jazyků kontextových. Vhodnou modifikací, která bude představena později, se pokusíme sílu jazyků E0L gramatik zvýšit na sílu turingových strojů. Nyní se budeme zabývat nejzákladnějšími 0L systémy, s jejichž pomocí budeme definovat E0L gramatiky.

Poznámka: další gramatiky založené na 0L systémech (ET0L, CT0L, FEP0L, atd.) a vztahy mezi nimi lze najít v [6] a [7].

3.1 0L systém

0L systém, nebo také *0L gramatika*, je trojice $G = (\Sigma, P, w)$, kde

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

3.2 D0L systém

Nechť $G = (\Sigma, P, w)$ je 0L systém. Pokud pro každé $a \in \Sigma$ existuje právě jedno pravidlo $a \rightarrow x \in P, \quad x \in \Sigma^*$. Potom G je *D0L systém* (D - "deterministic").

3.3 P0L systém

Nechť $G = (\Sigma, P, w)$ je 0L systém. Pokud pro každé pravidlo $a \rightarrow x \in P$ platí $x \neq \varepsilon$. Potom G je *P0L systém* (P - "propagating").

Definice 3.1 Nechť $G = (\Sigma, P, w)$ je 0L systém a u, v jsou řetězce ze Σ^*. Mezi řetězci u a v platí relace $\Rightarrow_G$, nazývaná *přímá derivace*, jestliže můžeme řetězce u a v vyjádřit ve tvaru:

$$u = a_1 a_2 \dots a_n$$
$$v = b_1 b_2 \dots b_n$$
$$a_i \in \Sigma, \quad b_i \in \Sigma^* \text{ a } a_i \rightarrow b_i \in P \text{ pro } 1 \leq i \leq n.$$

Platí-li mezi řetězci u a v relace přímé derivace, pak píšeme $u \Rightarrow_G v$ a říkáme, že řetězec v lze přímo generovat z řetězce u v gramatice (systému) G. Je-li z kontextu zřejmé, že jde o derivaci v gramatice G, pak nemusíme specifikaci gramatiky (systému) pod symbolem $\Rightarrow$ uvádět.

Definice 3.2 Nechť $G = (\Sigma, P, w)$ je 0L systém a u, v jsou řetězce ze Σ^*. Mezi řetězci u a v platí relace $\Rightarrow^+$ nazývaná *derivace*, jestliže existuje posloupnost přímých derivací $x_{i-1} \Rightarrow x_i$, $1 \leq i \leq n$ taková, že platí:

$$u = x_0 \Rightarrow x_1 \Rightarrow \ldots \Rightarrow x_{n-1} \Rightarrow x_n = v.$$

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

Definice 3.3 Jestliže v 0L systému G platí pro řetězce u a v relace $u \Rightarrow^+ v$ nebo identita $u = v$, pak píšeme $u \Rightarrow^* v$. Relace $\Rightarrow^*$ je tranzitivním a reflexivním uzávěrem relace přímé derivace $\Rightarrow$.

Definice 3.4 Nechť $G = (\Sigma, P, w)$ je 0L systém. Řetězec $u \in \Sigma^*$ nazýváme *větnou formou*, jestliže platí $w \Rightarrow^* u$, tj. řetězec u je generovatelný z axiomu w. V 0L systémech se každá větná forma nazývá *věta*. *Jazyk* $L(G)$, generovaný systémem G, je definován množinou všech vět

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

Příklad 3.1 Mějme 0L systém $G = (\{a\}, \{a \to aa\}, a)$. Tento 0L systém je zároveň PD0L systémem a přijímá jazyk $L(G) = \{a^{2^n} : n \geq 0\}$.

3.4 E0L systém

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

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

Definice 3.5 Nechť $G = (V, \Sigma, P, w)$ je E0L systém a nechť λ a μ jsou řetězce z V^*. Mezi řetězci λ a μ platí relace $\Rightarrow_G$, nazývaná *přímá derivace*, jestliže můžeme řetězce λ a μ vyjádřit ve tvaru

$$\lambda = x_1 x_2 \ldots x_n$$
$$\mu = y_1 y_2 \ldots y_n$$
$$\text{kde } x_i \in V, \quad y_i \in V^* \text{ a } x_i \to y_i \in P \text{ pro } 1 \leq i \leq n.$$

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

Definice 3.6 Nechť $G = (N, \Sigma, P, w)$ je E0L systém a λ a μ jsou řetězce z V^*. Mezi řetězci λ a μ platí relace $\Rightarrow^+$ nazývaná *derivace*, jestliže existuje posloupnost přímých derivací $v_{i-1} \Rightarrow v_i \quad 1 \leq i \leq n$ taková, že platí:

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

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

Definice 3.7 Jestliže v systému G platí pro řetězce λ a μ relace $\lambda \Rightarrow^{+} \mu$ nebo identita $\lambda = \mu$, pak píšeme $\lambda \Rightarrow^{*} \mu$. Relace $\Rightarrow^{*}$ je tranzitivním a reflexivním uzávěrem relace přímé derivace $\Rightarrow$.

Definice 3.8 Nechť $G = (V, \Sigma, P, w)$ je E0L systém. Řetězec $\alpha \in V^{*}$ nazýváme *větnou formou*, jestliže platí $S \Rightarrow^{*} \alpha$, tj. řetězec α je generovatelný z axiomu w. Větná forma, která obsahuje pouze terminální symboly, se nazývá *věta*. *Jazyk* $L(G)$, generovaný systémem G, je definován množinou všech vět

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

4 Úvod do základních algebraických struktur

Při definici pojmu E0L gramatika nad volnou grupou a bezkontextová gramatika nad volnou grupou se neobejdeme bez znalosti základních algebraických struktur. Jedná se zejména o *binární operace*, *grupoidy, pologrupy, monoidy* a *grupy.* Na konci této kapitoly uvedeme definici *volné grupy*, kterou poté spojíme s E0L gramatikou a gramatikou bezkontextovou. U nově vzniklých struktur budeme zkoumat jejich schopnosti generovat jazyky.

4.1 Finitární algebraické operace

Základním pojmem ve výše uvedených strukturách je pojem *algebraické operace.* Mějme dáno celé nezáporné číslo n a množinu M. Řekneme, že na množině M je definována $n-$ární algebraická operace $\circ$, když každé uspořádané $n-$tici prvků $a_1, a_2, \ldots, a_n \in M$ je přiřazen jednoznačně určený prvek z M. Tento prvek nazveme výsledkem operace $\circ$ na dané prvky a budeme jej označovat symbolem $a_1 a_2 \ldots a_n \circ$. Číslo n nazýváme *aritou* operace $\circ$.

Definice 4.1 *Nulární operací* na množině M nazýváme zobrazení $\circ : M^0 \rightarrow M$, jehož výsledkem je jeden z prvků množiny M, který není závislý na volbě prvků z M.

Tuto operaci můžeme chápat jako vyčlenění jednoho význačného prvku z množiny M.

Definice 4.2 *Unární operací* na množině M nazýváme zobrazení $\circ : M \rightarrow M$, které každému prvku $a \in M$ přiřazuje právě jeden prvek $a\circ \in M$.

Definice 4.3 *Binární operací* na množině M nazýváme zobrazení $\circ : M^2 \rightarrow M$. Prvek $a \circ b$, kde $a, b, a \circ b \in M$, nazýváme kompozicí prvků a, b vzhledem k binární operaci na množině M.

Podle základních požadavků kladených na binární operace na dané množině dostáváme objekty s různými algebraickými vlastnostmi. Tyto objekty nazýváme *algebraické struktury.* Binární operace na určité množině, které splňují axiómy příslušné algebraické struktury, nazýváme *operacemi této struktury.*

4.2 Axiomy binárních operací

Každý axiom budeme chápat jako požadavek, který na danou binární operaci klademe. Uvažujme binární operaci $\circ$ a množinu M. Nyní uveďme základní axiomy, které nás budou v souvislosti s naší problematikou zajímat.

- 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 $e \in M$, pro který platí $e \circ a = a \circ e = a$ 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 $a \circ \overline{a} = \overline{a} \circ a = e$, kde $e \in M$ je neutrální prvek operace $\circ$ na množině M podle axiomu A_2.

Poznamenejme, že existují další axiomy definující další vlastnosti kladené a operace. My si však v tomto textu vystačíme s výše uvedenými a další proto nebudeme uvádět.

4.3 Grupoidy, pologrupy, monoidy a grupy

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

Definice 4.4 Algebraická struktura $(M, \circ)$ definována binární operací $\circ$ na množině M se nazývá *grupoid*, pokud operace $\circ$ splňuje axiom A_0 uzavřenosti operace.

Definice 4.5 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*.

Definice 4.6 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 $\varepsilon \in M$.

Definice 4.7 Algebraická struktura $(M, \circ, \varepsilon, ^{-1})$, která splňuje axiomy A_0, A_1, A_2 a A_3, se nazývá *grupa.*

Grupu můžeme chápat jako monoid obsahující ke každému prvku $a \in M$ prvek inverzní $\overline{a} \in M$, kde jeho přiřazení každému prvku můžeme provést definicí unární operace $^{-1} : M \to M$.

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.

5 Volné grupy

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

5.1 Volný monoid

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

Definujme nyní výše uvedené formálně.

Definice 5.1 Množina V se nazývá *volná báze* monoidu G, jestliže V generuje G a každé zobrazení $f : V \to H$, kde H je monoid, lze rozšířit na homomorfismus $g : G \to H$. Monoid G se nazývá *volný*, jestliže má alespoň jednu volnou bázi.

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

5.2 Volná grupa

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

Uveďme si formální definici.

Definice 5.2 Množina V se nazývá *volná báze* grupy G, jestliže V generuje G a každé zobrazení $f : V \to H$, kde H je grupa, lze rozšířit na homomorfismus $g : G \to H$. Grupa G se nazývá *volná*, jestliže má (alespoň jednu) volnou bázi.

Aby nedošlo k záměně označení volného monoidu a volné grupy, stanovme si zde, že volnou grupu generovanou množinou generátorů V pomocí operace konkatenace budeme značit jednoduše symbolem V°. Protože množina V obsahuje ke každému prvku i prvek inverzní, mohou se všechny prvky z V vyskytovat v jednotlivých slovech množiny V°. Každé slovo, které obsahuje dvojice $\overline{x}x$ nebo $x\overline{x}$, lze dále redukovat až k jejich úplnému odstranění. Slovo, které neobsahuje žádné výše uvedené dvojice prvků vzájemně inverzních, se potom nazývá *redukované*. Lze dokázat, že každému slovu odpovídá pouze jediné slovo redukované a to bez ohledu na pořadí redukcí jednotlivých dvojic vzájemně inverzních symbolů. Důkaz je možné najít např. v [4].

V pozdějším spojení gramatiky a volné grupy však budeme možnost redukce dvojic vzájemně inverzních symbolů hojně využívat.

6 Gramatiky nad volnými grupami

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

Vzhledem k zadanému rozsahu této práce nebudou uváděny důkazy ekvivalencí tříd jazyků rekurzivně vyčíslitelných s jazyky přijímanými gramatikami bezkontextovými nad volnými grupami a s jazyky přijímanými E0L gramatikami nad volnými grupami.

6.1 Bezkontextové gramatiky nad volnými grupami

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

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

Definice 6.1 *Bezkontextová gramatika nad volnou grupou* (zkráceně $\mathbf{CF}^\circ$) je čtveřice $\Gamma = (V, \Sigma, P, S)$, kde V, Σ a S má stejný význam jako v gramatikách bezkontextových. Dále P je konečná množina pravidel tvaru $A \to x$, kde $A \in V - \Sigma$ a $x \in V^\circ$.

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

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

6.2 Derivace v bezkontextové gramatice nad volnou grupou

Definice 6.2 Nechť $\Gamma = (V, \Sigma, P, S)$ je bezkontextová gramatika nad volnou grupou. Mezi řetězci λ a μ platí relace $\Rightarrow^\circ$ zvaná přímá derivace, pokud oba řetězce můžeme vyjádřit ve tvaru

$$\lambda = \alpha A \beta$$
$$\mu = \alpha x \beta$$

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

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

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

6.3 Jazyk generovaný bezkontextovou gramatikou nad volnou grupou

Definice 6.3 Nechť $\Gamma = (V, \Sigma, P, S)$ je bezkontextová gramatika nad volnou grupou. Řetězec $\alpha \in V^\circ$ nazýváme *větnou formou*, jestliže platí $S \Rightarrow^{\circ *} \alpha$, tj. řetězec α je generovatelný ze startovacího symbolu S. Větná forma, která obsahuje pouze terminální symboly, se nazývá *věta*. *Jazyk* $L(\Gamma)$, generovaný bezkontextovou gramatikou nad volnou grupou, je definován množinou všech vět

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

generovatelných z výchozího symbolu této gramatiky.

6.4 Generativní schopnosti bezkontextových gramatik nad volnými grupami

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

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

Pravidla tvaru $A \to \varepsilon$, která mohou být aplikována kdekoliv, působí potíže při práci s volnou grupou (konkrétně se jedná o simulaci kontextových pravidel), proto byla z [2] převzata následující pomocná věta. Ta zajistí převedení pravidel tvaru $A \to \varepsilon$ na pravidla tvaru $A \to B$. Ke gramatice je poté přidána určitá režie, která zajistí přijímání původního jazyka. V této režii se vyskytuje pouze jedno pravidlo tvaru $A \to \varepsilon$, jeho použití je snadné ošetřit a ve výsledné gramatice nad volnou grupou nečiní jeho výskyt potíže. Nyní si uvedeme zmíněnou větu.

Věta 6.1 Pro každou gramatiku $H = (N, \Sigma, P, S_H)$ typu 0 existuje ekvivalentní gramatika $G = (N \cup \{X, Y\}, \Sigma, P, S_G)$ typu 0, kde $\{X, Y, S_G\} \cap \{N \cup \Sigma\} = \emptyset$ taková, že její množina přepisovacích pravidel obsahuje pouze pravidla tvaru:

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

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

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

(4) $S_G \to XS_H$,

(5) $XY \to X$

(6) $X \to \varepsilon$

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

Důkaz 6.1 Z důkazu si uvedeme pouze konstrukci, kterou je též možné najít v [2]. Uvažujme gramatiku $H = (N_H, \Sigma, P_H, S_H)$. Bez ztráty obecnosti můžeme předpokládat, že H je v Kurodově normální formě, tedy množina přepisovacích pravidel P_H obsahuje pouze pravidla tvaru

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

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

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

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

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

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

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

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

Tím je konstrukce hotova.

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

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

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

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

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

Věta 6.2 **CF**$^\circ$=**RE**

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

Důkaz 6.2 Z důkazu bude prezentována pouze konstrukce.

Předpokládejme, že $G = (N_G \cup \{X, Y\}, \Sigma, P_G, S)$ je gramatika typu 0 podle *věty* 6.1.

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

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

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

III pokud $AY \to YA \in P_G$,
přidej $A \to Y\langle AY \rangle$,
$Y \to \langle \overline{AY} \rangle A$ do P_Γ
a $\langle AY \rangle, \langle \overline{AY} \rangle$ přidej do N_{CS}

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

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

VI pro $Z \in N_G \cup \{X, Y\} \cup \Sigma$
přidej $\overline{Z}$ do $\overline{N_G}$

Tím je konstrukce Γ hotova.

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

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

6.5 E0L gramatiky nad volnými grupami

Druhá modifikace se bude týkat E0L gramatik. Konstrukce bude velmi podobná, proto bude popsána stručněji. Důvodem proč byly zvoleny E0L gramatiky, je jejich paralelní přístup k derivování větných forem a dále jsme jejich pomocí získali lepší výsledky v oblasti redukce nonterminálů (viz. [11]).

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

Definice 6.4 *E0L gramatika nad volnou grupou* (zkráceně **E0L**$^\circ$) je čtveřice $\Gamma = (V, \Sigma, P, w)$, kde V a Σ má stejný význam jako v E0L gramatikách. Dále $w \in V^\circ$ a P je konečná množina pravidel tvaru $X \to x$, kde $X \in V$ a $x \in V^\circ$.

Pro úplnost definujeme relaci přímé derivace $\Rightarrow^\circ_\Gamma$.

6.6 Derivace v E0L gramatice nad volnou grupou

Definice 6.5 Nechť $\Gamma = (V, \Sigma, P, w)$ je E0L gramatika nad volnou grupou, kde $V = N \cup \Sigma$. Nechť λ a μ jsou řetězce z V°. Mezi řetězci λ a μ platí relace $\Rightarrow^\circ_\Gamma$, nazývaná *přímá derivace*, jestliže můžeme řetězce λ a μ vyjádřit ve tvaru:

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

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

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

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

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

6.7 Jazyk generovaný E0L gramatikou nad volnou grupou

Definice 6.6 Nechť $\Gamma = (V, \Sigma, P, w)$ je E0L gramatika nad volnou grupou, kde $V = N \cup \Sigma$. Řetězec $\alpha \in V^\circ$ nazýváme *větnou formou*, jestliže platí $w \Rightarrow^* \alpha$, tj. řetězec α je generovatelný z výchozího řetězce w. Větná forma, která obsahuje pouze terminální symboly, se nazývá *věta*. *Jazyk* $L(\Gamma)$, generovaný E0L gramatikou nad volnou grupou, je definován množinou všech vět

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

generovatelných z výchozího řetězce této gramatiky.

6.8 Generativní schopnosti E0L gramatik nad volnými grupami

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

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

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

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

Věta 6.3 **E0L**$^\circ$=**RE** Jinými slovy tato věta říká, že pro každou gramatiku G typu 0 existuje ekvivalentní E0L gramatika nad volnou grupou Γ taková, že $L(G) = L(\Gamma)$.

Důkaz 6.3 Z důkazu bude prezentována pouze konstrukce.

Předpokládejme, že $G = (N_G \cup \{X, Y\}, \Sigma, P_G, S)$ je gramatika typu 0 podle *věty* 6.1.

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

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

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

III pokud $AY \to YA \in P_G$,
přidej $A \to Y\langle AY \rangle$,
$Y \to \langle \overline{AY} \rangle A$ do P_Γ
a $\langle AY \rangle, \langle \overline{AY} \rangle$ přidej do N_{CS}

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

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

VI pro $Z \in N_G \cup \{X, Y\} \cup \Sigma$
přidej $\overline{Z}$ do $\overline{N_G}$

VII pro každé $A \in V \cup \{X, Y\}$
přidej $A \to A$ do P_Γ

Tím je konstrukce Γ hotova.

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

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

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

7 Závěr

V tomto dokumentu jsme zavedli nové modely umožňující generovat jazyky. Pomocí těchto modelů získáváme sílu Turingova stroje, a to přidáním volné grupy k bezkontextové nebo E0L gramatice, tím výrazně zvyšujeme jejich generativní schopnosti.

Nevýhodou je nárůst nonterminálních symbolů a pravidel. Další vývoj tohoto projektu směřuje k redukci nonterminálních symbolů, viz [11].

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

Reference

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

[2] Meduna, A., Kolář, D.: Homogenous Grammars with a Reduced Number of Non-Context-Free Productions. In: Information Processing Letters, roč. 2002, č. 81, Amsterdam, NL, p. 253-257, ISSN 0020-0190.

[3] Penttonen, M.: One-Sided and Two-Sided Context in Formal Grammars. Inf. Control 25, pp. 371-392, 1974.

[4] Milne, J.S.: Group Theory. Course Notes, 2003, [http://www.jmilne.org/math/CourseNotes/math594g.pdf].

[5] Meduna, A.: Context Free Derivations on Word Monoids. Acta Informatica, Springer-Verlag, 1990.

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

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

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

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

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

[11] Blatný, P., Bidlo, R.: The Parallel Generation of Recursively Enumerable Languages Using Only Context-free Productions and Six Nonterminals. Brno, EEICT 2005, Vol 3, nakl. VUT, 2005.