String-Partitioning Systems

Projekt do předmětu TID

Rudolf Schönecker
schonec@fit.vutbr.cz

Abstrakt Řetězce rozdělující systémy představují model založený na principu přepisování použitím gramatických pravidel. V systému neexistují nonterminály tak, jak je známe z klasických gramatik Chomského klasifikace, k dispozici je pouze jeden speciální symbol, tzv. bounder, který udává ve větné formě pozici, kam bude vkládána jiná větná forma. Omezíme-li systém konečným indexem, získáváme nekonečnou hierarchii tříd generovaných jazyků. Práce shrnuje potřebné prerekvizity, uvádí důkaz ekvivalence s programovanými gramatikami s konečným indexem a existence nekonečné hierarchie řetězce rozdělujících systémů s konečným indexem.

Klíčová slova: teorie formálních jazyků, konečný index, řetězce rozdělující systémy, programované gramatiky, syntaktická analýza, překladače

1 Úvod

Přepisovací systémy stály již od počátku v teorii formálních jazyků jako kompromis popisu jazyka mezi srozumitelným – pro člověka – a zároveň snadno algoritmicky aplikovatelným – z pohledu informačních technologií. Byly vyvinuty různé typy přepisovacích systémů, následně kategorizovány, vzájemně porovnávány. Rozhodující byly vždy klíčové vlastnosti, z hlediska jazyků se jednalo o popisnou sílu systémů a potažmo jazyků, resp. tříd jazyků jimi generovaných.

Postupně byly vymezovány hranice mezi jednotlivými systémy a třídami jazyků, specifikováno jejich užití na konkrétní aplikace, zejména v informatice, moderní biologii a genetice. Základní aparát gramatik a automatů na úrovni regulárních a bezkontextových jazyků byl často využíván pro svou jednoduchost konstrukce a zároveň dostatečnou sílu z hlediska požadovaných vlastností systémů a zdá se dostačujícím např. v oblasti překladačů a algoritmizace.

Z hlediska teorie formálních jazyků se striktně oddělují gramatiky a automaty jako systémy, které generují resp. přijímají jazyky. V současnosti je však potřeba systémů, které zahrnují některé vlastnosti obou. Do této kategorie můžeme zařadit i řetězce rozdělující systémy, které na jedné straně poskytují formalismus přepisu podobný gramatikám, na druhé straně však neobsahují nonterminální symboly, pouze jeden speciální symbol pro lokaci pozice dalších přepisování, navíc jsou opatřeny stavy, čímž zase inklinují k automatům.

Aplikace podobných systémů můžeme hledat především v mikrobiologii (simulace buněčných organismů) či makrobiologii (simulace práce s genetickou informací a genetickými strukturami v organismech), experimentální použití např. v syntaktické analýze.

Úvodní poznámka
Tato práce vychází z již existujících materiálů o řetězce rozdělujících systémech, zahrnuje však i části připravovaného článku o zařazení řetězce rozdělujících systémů s konečným indexem do kontextu řízených gramatik s konečným indexem. Tyto partie jsou z časových důvodů ponechány v anglickém jazyce.

2 Prerekvizity

Mějme abecedu V. Pro $w \in V^*$, $|w|$ je používáno ve významu délky řetězce w, a pro $W \subseteq V$, $occur(w, W)$ reprezentuje počet výskytů symbolů z W v w.

2.1 Programované gramatiky

Programovaná gramatika je čtveřice $G = (V, T, P, S)$, kde V je celková abeceda, $T \subseteq V$ je abeceda terminálních symbolů, $S \in (V - T)$ je startovací symbol, a P je konečná množina pravidel ve tvaru $q\colon A \to v, g(q)$, kde $q\colon A \to v$ je bezkontextové pravidlo označené symbolem q a $g(q)$ představuje množinu symbolů označujících pravidla asociovaných s pravidlem q. Po aplikaci pravidla q v daném tvaru na libovolný nonterminál ve větné formě mající stejný název jako nonterminál na levé straně pravidla, bude v dalším kroku aplikováno jedno z pravidel z množiny $g(q)$ v případě, že je $g(q)$ neprázdná, jinak se derivace v gramatice zablokuje. Derivační krok označujeme podobně jako u bezkontextových gramatik symbolem $\Rightarrow$, podobně také $\Rightarrow^m$, kde $m \geq 0$, $\Rightarrow^+$, a $\Rightarrow^*$. Jazyk generovaný programovanou gramatikou G značíme $L(G)$, je definován jako $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$.

Programované gramatiky s testováním výskytu (with appearance checking) jsou programované gramatiky, kdy pro každé pravidlo q z množiny P je definována navíc množina $r(q)$ symbolů označující pravidla asociovaná s pravidlem q. Množina $g(q)$ bývá nazývána *pole úspěchu* a množina $r(q)$ *pole neúspěchu*. Pravidlo programované gramatiky lze poté použít dvěma způsoby. Pokud možnost spočívá v klasické aplikaci pravidla, tj. nahrazení nonterminálu, tvořícího levou stranu pravidla, pravou stranou pravidla. Pokud má právidlo neprázdné pole neúspěchu, lze jej aplikovat ve smyslu testování výskytu: Není-li levá strana pravidla obsažena ve slově, na které pravidlo používáme, zůstane slovo beze změny. V dalším kroku musíme užít některé z pravidel v poli neúspěchu.

Příklad: $L = \{a^n b^m c^n d^m \mid n, m \geq 1\}$
$G = (\{S, A, B, C, D\}, \{a, b, c, d\}, \{f_1, \ldots, f_9\}, S)$

- $f_1 = (S \to ABCD, \{f_2, f_3, f_6\}, \emptyset)$
- $f_2 = (A \to aA, \{f_4\}, \emptyset)$
- $f_3 = (B \to bB, \{f_5\}, \emptyset)$
- $f_4 = (C \to cC, \{f_2, f_3, f_6\}, \emptyset)$
- $f_5 = (D \to dD, \{f_2, f_3, f_6\}, \emptyset)$
- $f_6 = (A \to a, \{f_7\}, \emptyset)$
- $f_7 = (B \to b, \{f_8\}, \emptyset)$
- $f_8 = (C \to c, \{f_9\}, \emptyset)$
- $f_9 = (D \to d, \emptyset, \emptyset)$

2.2 Konečný index gramatik

Nyní se zaměříme na pojem *gramatik konečného indexu*. Neformálně řečeno, index gramatiky je maximální počet neterminálů, které se během derivací mohou současně vyskytovat v jedné větné formě. Konečnost indexu je velice přirozená a silná podmínka, která významně slejdnocuje sílů různých typů gramatik.

Nechť G je gramatika libovolného typu a V_N, V_T jsou její neterminální, resp. terminální abeceda a S startovací symbol. Pro derivaci $D\colon S = w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_r = w \in V_T^*$, podle G definujeme index D jako

$$Ind(D, G) = \max\{occur(w_i, V_N) \mid 1 \leq i \leq r\},$$

a pro $w \in V_T^*$ definujeme index w jako

$$Ind(w, G) = \min\{Ind(D, G) \mid D\text{je derivace } w \text{ v } G\}.$$

Index gramatiky G je definován jako

$$Ind(G) = \sup\{Ind(w, G) \mid w \in L(G)\}.$$

Pro jazyk L ze třídy jazyků $\mathcal{L}(X)$ generovaných gramatikami typu X definujeme index jazyka jako

$$Ind_X(L) = \inf\{Ind(G) \mid L(G) = L, \text{kde } G \text{ je typu } X\}.$$

Pro třídu jazyků $\mathcal{L}(X)$ definujeme $\mathcal{L}_n(X) = \{L \mid L \in \mathcal{L}(X) \text{ and } Ind_X(L) \leq n\}$, $n \geq 1$ a $\mathcal{L}_{fin}(X) = \bigcup_{n \geq 1} L_n(X)$.

2.3 Známé výsledky o generativní síle vybraných gramatik s konečným indexem

Pro jazyky indexu n pro všechna $n \geq 1$ platí

$$\mathcal{L}_n(M, CF - \lambda) \subseteq \mathcal{L}_n(M, CF)$$

$$\mathcal{L}_n(M, CF - \lambda, ac) \subseteq \mathcal{L}_n(M, CF, ac),$$

tj. použití λ-pravidel zvyšuje sílu jazyka s indexem n. Navíc pro všechna $X \in \{CF, CF - \lambda\}$ platí

$$\mathcal{L}_n(M, X) \subseteq \mathcal{L}_n(M, X, ac)$$

tj. použití testování výskytu opět zvyšuje sílu jazyka s indexem n.

Pro třídy jazyků s konečným indexem platí

$$\mathcal{L}_{fin}(M, CF) = \mathcal{L}_{fin}(M, CF - \lambda) = \mathcal{L}_{fin}(M, CF, ac) = \mathcal{L}_{fin}(M, CF - \lambda, ac),$$

což znamená, že u zmíněných řízených gramatik s konečným indexem nezáleží na použití λ-pravidel či testování výskytu, jejich generativní síla zůstane nezměněna.

Všechny následující třídy jazyků k konečným indexem jsou ekvivalentní pro $X \in \{CF, CF - \lambda\}$ jsou ekvivalentní s třídou $\mathcal{L}_{fin}(M, CF)$: $\mathcal{L}_{fin}(P, X)$, $\mathcal{L}_{fin}(P, X, ac)$, $\mathcal{L}_{fin}(RC, X, ac)$.

Dále platí nekonečná hierarchie pro každý pro všechna $n \geq 1$

$$\mathcal{L}_n(M, CF) \subset \mathcal{L}_{n+1}(M, CF).$$

Pro každý nekonečný jazyk $L \in \mathcal{L}_n(M, CF)$ existuje řetězec $z \in L$, který lze zapsat ve tvaru: $z = u_1 v_1 w_1 x_1 u_2 v_2 w_2 x_2 \ldots u_k v_k w_k x_k u_{k+1}$, kde $k \leq n$, $|v_1 x_1 v_2 x_2 \ldots v_k x_k| > 0$ a pro všechna $i \geq 1$ platí $u_1 v_1^i w_1 x_1^i u_2 v_2^i w_2 x_2^i \ldots u_k v_k^i w_k x_k^i u_{k+1} \in L$.

Poslední poznámka směřuje ke třídě bezkontextových jazyků a třídě jazyků s konečným indexem generovaných maticovými gramatikami. Platí, že tyto dvě třídy jsou navzájem neporovnatelné, protože

$$\mathcal{L}(CF) - \mathcal{L}_{fin}(M, CF) \neq \emptyset.$$

3 Definice

Nechť I je konečná množina přirozených čísel $\{1, 2, \ldots, k\}$. Řetězce rozdělující systém (SPS) je čtveřice $M = (Q, \Sigma, s, R)$, kde Q je konečná množina stavů, Σ je abeceda obsahující speciální symbol, $\#$, nazývaný značka (bounder), $s \in Q$ je počáteční stav, a $R \subseteq Q \times I \times \{\#\} \times Q \times \Sigma^*$ je konečná relace, jejíž prvky nazýváme pravidla. Pravidlo $(q, n, \#, p, x) \in R$, kde $n \in I$, $q, p \in Q$ a $x \in \Sigma^*$, se zapisuje ve tvaru $q\,{}_n\# \rightarrow px$.

K-limitovaná konfigurace je řetězec $x \in Q\Sigma^*$, pro který platí $occur(x, \#) \leq k$. Nechť $pu\#v$, $quxv$ jsou dvě k-limitované konfigurace $u, v \in \Sigma^*$, $occur(u, \#) = n - 1$ a $p\,{}_n\# \rightarrow qx \in R$. Pak

1. M provede *derivační krok* z konfigurace $pu\#v$ do $quxv$ použitím pravidla $p\,{}_n\# \rightarrow qx$, symbolicky zapíšeme $pu\#v\ {}_d\Rightarrow\ quxv\ [p\,{}_n\# \rightarrow qx]$ v M
2. M provede *redukční krok* z konfigurace $quxv$ do $pu\#v$ použitím pravidla $p\,{}_n\# \rightarrow qx$, symbolicky zapíšeme $quxv\ {}_r\Rightarrow\ pu\#v[p\,{}_n\# \rightarrow qx]$ v M.

Nechť ${}_d\Rightarrow^*$ a ${}_r\Rightarrow^*$ znamenají tranzitivní a reflexivní uzávěr relace ${}_d\Rightarrow$ resp. ${}_r\Rightarrow$.

Jazyk derivovaný řetězce rozdělujícím systémem M, $L(M, {}_d\Rightarrow)$, je definovaný jako

$$L(M, {}_d\Rightarrow) = \{w \mid s\#\ {}_d\Rightarrow^*\ qw,\ q \in Q, w \in (\Sigma - \{\#\})^*\}.$$

Jazyk redukovaný řetězce rozdělujícím systémem M, $L(M, {}_r\Rightarrow)$, je definovaný jako

$$L(M, {}_r\Rightarrow) = \{w \mid qw\ {}_r\Rightarrow^*\ s\#,\ q \in Q, w \in (\Sigma - \{\#\})^*\}.$$

Příklad:
$M = (\{s, p, q, f\}, \{a, b, c, \#\}, s, R)$, kde R obsahuje tato pravidla:

1. $s\,{}_1\# \rightarrow p\ \#\#$
2. $p\,{}_1\# \rightarrow q\ a\#b$
3. $q\,{}_2\# \rightarrow p\ \#c$
4. $p\,{}_1\# \rightarrow f\ ab$
5. $f\,{}_1\# \rightarrow f\ c$

$L(M, {}_d\Rightarrow) = \{a^n b^n c^n \mid n \geq 1\} = L(M, {}_r\Rightarrow)$ splňuje $Ind(M) = 2$.

Příklad derivace řetězce $aaabbbccc$:
$s\#\ {}_d\Rightarrow\ p\#\#[1]\ {}_d\Rightarrow\ qa\#b\#\ [2]\ {}_d\Rightarrow\ pa\#b\#c\ [3]\ {}_d\Rightarrow\ qaa\#bb\#c\ [2]\ {}_d\Rightarrow\ paa\#bb\#cc\ [3]\ {}_d\Rightarrow$
$faaabbb\#cc\ [4]\ {}_d\Rightarrow\ faaabbbccc\ [5]$.

Příklad redukce řetězce $aaabbbccc$:
$faaabbbccc\ {}_r\Rightarrow\ faaabbb\#cc\ [5]\ {}_r\Rightarrow\ paa\#bb\#cc[4]\ {}_r\Rightarrow\ qaa\#bb\#c\ [3]\ {}_r\Rightarrow\ pa\#b\#c\ [2]\ {}_r\Rightarrow$
$qa\#b\#\ [3]\ {}_r\Rightarrow\ p\#\#\ [2]\ {}_r\Rightarrow\ s\#\ [1]$.

Let $\mathcal{L}_{fin}(SPS, {}_d\Rightarrow)$, and $\mathcal{L}_{fin}(P, CF)$ denote the families of string-partitioning system derived languages, and programmed languages of finite index based on context-free grammar, respectively.

4 Výsledky

4.1 Ekvivalence s programovanými gramatikami s konečným indexem

Lemma 1. $\mathcal{L}_k(P, CF) \subseteq \mathcal{L}_k(SPS, {}_d\Rightarrow)$
For every programmed grammar of index k, G, there is a string-partitioning system of index k, H, such that $L_k(G) = L_k(H, {}_d\Rightarrow)$.

Důkaz Lemma 1 je tvořen dvěma částmi: konstrukční a indukční. Konstrukční část důkazu popisuje způsob, jakým lze pro každou programovanou gramatiku vytvořit ekvivalentní řetězce rozdělující systém. Vychází z myšlenky kódovat použití následujícího pravidla (tak jak je specifikováno v programované gramatice v poli úspěchu) přímo do názvu stavu řetězce rozdělujícího systému. Jestliže máme tedy např. konfiguraci $\langle A_1A_2 \dots A_{j-1}A_jA_{j+1} \dots A_h\rangle x_0\#x_1\# \dots \#x_n$, kde počet značek je n, a v následujícím kroku má být přepsána j-tá značka pomocí pravidla p, vyznačíme tuto skutečnost přímo do názvu stavu jako $\langle A_1A_2 \dots A_{j-1}[p]A_{j+1} \dots A_h\rangle$. Je nutné rovněž ošetřit blokující konfigurace v programované gramatice, tj. konfigurace ve kterých neexistuje již žádné pravidlo pro aplikaci, řetězce rozdělující systém musí mít ekvivalentní možnosti vyjádření této situace.

Indukční část důkazu pak potvrzuje správnost konstrukce řetězce rozdělujícího systému. Dokazuje, že pro každou derivaci v programované gramatice s indexem k existuje odpovídající derivace v řetězce rozdělujícím systému s indexem k s ohledem na příslušný postup použitý při jeho konstrukci na základě dané gramatiky. První tvrzení vyslovuje obecnou větu o existenci dané derivace, následuje jeho důkaz, druhé tvrzení je specializací prvního tvrzení a v důsledku uvedeného důkazu prvního tvrzení je rovněž platné.

Proof. Let $k \geq 1$ be a positive integer. Let $G = (V, T, P, S)$ is programmed grammar of index k, where $N = V - T$. We construct the string-partitioning system of index k, $H = (Q, T \cup \{\#\}, s, R)$, where $\# \notin T$, $s = \langle\sigma\rangle$, σ is a new symbol, R and Q are constructed by performing the following steps:

1. For each $p\colon S \to \alpha \in P$, $\alpha \in V^*$, add $\langle\sigma\rangle_1\# \to \langle[p]\rangle\#$ to R, $\langle[p]\rangle$ is new state in Q.
2. If $A_1A_2 \dots A_j \dots A_h \in N^*$, $h \in \{1, 2, \dots, k\}$, $p\colon A_j \to x_0B_1x_1B_2x_2 \dots x_{n-1}B_nx_n$, $g(p) \in P$, $j \in \{1, 2, \dots, h\}$ for $n \geq 0$, $x_t \in T^*$, $B_t \in N$, $0 \leq t \leq n$ and $n + h - 1 \leq k$, then
   (a) if $g(p) = \emptyset$, then $\langle A_1A_2 \dots A_{j-1}[p]A_{j+1} \dots A_h\rangle$, $\langle A_1A_2 \dots B_1 \dots B_n \dots A_h\rangle$ are new states in Q and the rule $\langle A_1A_2 \dots A_{j-1}[p]A_{j+1} \dots A_h\rangle_j\# \to \langle A_1A_2 \dots B_1 \dots B_n \dots A_h\rangle\ x_0\# \dots \#x_n$ is added to R
   (b) for every $q \in g(p)$, $q\colon D_d \to \alpha$, $\alpha \in V^*$ add new states $\langle A_1A_2 \dots A_{j-1}[p]A_{j+1} \dots A_h\rangle$ and $\langle D_1D_2 \dots [q] \dots D_{n+h-1}\rangle$ to Q and add the following rule to R:
   $\langle A_1A_2 \dots A_{j-1}[p]A_{j+1} \dots A_h\rangle_j\# \to \langle D_1D_2 \dots [q] \dots D_{n+h-1}\rangle x_0\#x_1 \dots x_{n-1}\#x_n$, where $A_1 \dots A_{j-1}B_1 \dots B_nA_{j+1} \dots A_h = D_1 \dots D_{j-1}D_j \dots D_{j+n-1}D_{j+n} \dots D_{h+n-1}$, $B_1 \dots B_n = D_j \dots D_{j+n-1}$ for some $1 \leq d \leq n + h - 1$.

Claim 1. If $S \Rightarrow^m x_0A_1x_1A_2x_2 \dots x_{n-1}A_nx_n$ in G, then $\langle\sigma\rangle\#\ {}_d\Rightarrow^r\ \langle A_1A_2 \dots A_n\rangle x_0\#x_1 \dots x_n\ [q_1q_2 \dots q_r]$ in H, for $m \geq 1$. If $g(q_r) \neq \emptyset$, then exist a rule $q_{r+1}\colon A_j \to y_0B_1y_1 \dots y_{h-1}B_ny_n$, $n + h - 1 \leq k$, $q_{r+1} \in g(q_r)$ and $A_j = [q_{r+1}]$.

Basis: Let $m = 0$. For $S \Rightarrow^0 S$ in G there exists $\langle\sigma\rangle\#\ {}_d\Rightarrow^1\ \langle[p]\rangle\#$ in H, where $p\colon S \to \alpha \in P$ and $\langle\sigma\rangle_1\#\ {}_d\Rightarrow\ \langle[p]\rangle\# \in R$.

Induction Hypothesis: Suppose that Claim 1 holds for all derivations of length m or less for some $m \geq 0$.

Induction Step: Consider $S \Rightarrow^m y\ [p_1p_2 \dots p_m]$, where $y = x_0A_1x_1 \dots x_{n-1}A_nx_n$ and $p_{m+1} \in P$ so

that $y \Rightarrow x\ [p_{m+1}]$. If $m = 0$, then $p_{m+1} \in \{p \mid lhs(p) = S, p \in P\}$ otherwise $p_{m+1} \in g(p_m)$. For $p_{m+1}\colon A_j \rightarrow y_0B_1y_1 \dots y_{h-1}B_hy_h$ is x in the form:
$x = x_0A_1x_1 \dots A_{j-1}x_{j-1}y_0B_1y_1 \dots y_{h-1}\ B_hy_hx_jA_{j+1} \dots x_{n-1}A_nx_n$, for $x_0, \dots, x_n \in T^*$ and $y_0, \dots, y_h \in T^*$. Based on the induction hypothesis, there exists
$\langle\sigma\rangle\#\ {}_d\Rightarrow^*\ \langle A_1A_2 \dots A_{j-1}[p_{m+1}]A_{j+1} \dots A_n\rangle x_0\#x \dots \#x_n\ [q_1q_2 \dots q_r]\ {}_d\Rightarrow$
$\langle A_1A_2 \dots A_{j-1}B_1 \dots B_hA_{j+1} \dots A_n\rangle x_0\# \dots \#x_{j-1}y_0\# \dots \#y_hx_j\# \dots \#x_n\ [q_{r+1}]$, $q_i \in R$, $r \geq 1$, $1 \leq i \leq r+1$. If $g(p_{m+1}) \neq \emptyset$, then exists a rule $p_{m+2} \in g(p_{m+1})$ and a sequence $D_1D_2 \dots D_{n+h-1}$ so that $A_1A_2 \dots A_{j-1}B_1 \dots B_hA_{j+1} \dots A_h = D_1D_2 \dots D_{n+h-1}$ so that for some $d \in \{1, 2, \dots, n+h-1\}$ is $D_d = [q_{r+2}\colon A_d \rightarrow \gamma]$.

Claim 2. If $S \Rightarrow^z x$ in G, then $\langle\sigma\rangle\#\ {}_d\Rightarrow^*\ \langle\rangle x$ in H for some $z \geq 0$, $x \in T^*$.
Consider Claim 1 for $n = 0$. At this point, if $S \Rightarrow^z x_0$, then $\langle\sigma\rangle\#\ {}_d\Rightarrow^*\ \langle\rangle x_0$ and so $x_0 = x$. □

Lemma 2. $\mathcal{L}_k(SPS,\ {}_d\Rightarrow) \subseteq \mathcal{L}_k(P, CF)$
For every string-partitioning system of index k, H, exists equivalent programmed grammar of index k, G, such that $L_k(G) = L_k(H,\ {}_d\Rightarrow)$.

Důkaz Lemma 2 má opět dvě části. Konstrukční část důkazu nyní popisuje algoritmus konstrukce programované gramatiky s indexem k na základě existujícího řetězce rozdělujícího systému se stejným indexem. Opírá se přitom o speciálně nazvané neterminály gramatiky – jejich název tvoří vždy trojici a umožňuje takto v sobě nést informace o aktuálním "stavu", pořadí sama sebe ve větné formě, a celkový počet neterminálnů ve větné formě tak, aby byly k dispozici při zpracovávání libovolného neterminálu. Pro takto formované neterminály předepisuje pravidla, která při přepisu v gramatice navíc zajišťují aktualizaci "uložených" informací.

Indukční část důkazu pak potvrzuje správnost konstrukce řetězce rozdělujícího systému. Dokazuje, že pro každou derivaci v řetězce rozdělujícím systému s indexem k existuje odpovídající derivace v programované gramatice s indexem k s ohledem na příslušný postup použitý při její konstrukci na základě dané systému. První tvrzení opět vyslovuje obecnou větu o existenci dané derivace, následuje jeho důkaz, druhé tvrzení je specializací prvního tvrzení a v důsledku uvedeného důkazu prvního tvrzení je rovněž platné. Důkaz indukcí vychází z existence derivace délky 0. Ta samozřemě existuje v obou systémech. Následně předpokládá, že v řetězce rozdělujícím systému existuje derivace $\langle\sigma\rangle\#\ {}_d\Rightarrow^c \langle\vartheta\rangle y_0\#y_1 \dots y_{h-1}\#y_h\ [r_1r_2 \dots r_c]$ v H, a na výslednou formu $\langle\vartheta\rangle y_0\#y_1 \dots y_{h-1}\#y_h\ r_1r_2 \dots r_c]$ je možné aplikovat pravidlo r_{c+1}. Tuto skutečnost se pro analogické větné formy v gramatice snaží simulovat s použitím příslušných konstrukcí definovaných v konstrukční části důkazu.

Proof. Let $k \geq 1$ be a positive integer. Let $H = (Q, T \cup \{\#\}, s, R)$ is string-partitioning system of index k, where $\Sigma = T \cup \{\#\}$. We construct the programmed grammar of index k, $G = (V, T, P, S)$, and the sets of nonterminals $N = V - T$ and rules P are constructed as follows:

1. $P = \emptyset$,
2. $S = \langle s, 1, 1\rangle$,
3. $N = \{\langle p, i, h\rangle \mid p \in Q, 1 \leq i \leq k, 1 \leq h \leq k\} \cup \{\langle q', i, h\rangle \mid q \in Q, 1 \leq i \leq k, 1 \leq h \leq k\}$
   $\cup\ \{\langle q'', i, h\rangle \mid q \in Q, 1 \leq i \leq k, 1 \leq h \leq k\}$,
4. For every rule $r\colon p\ \#\rightarrow qy \in R$, $y = y_0\#y_1\#y_2 \dots y_{m-1}\#y_m$, $y_0, y_1, y_2 \dots y_m \in T^*$,
   if $m = 0$, then $h_{max} = k$ else $h_{max} = k - m + 1$, add following set to P:
   (i) $\{\langle p, j, h\rangle \rightarrow \langle q', j, h + m - 1\rangle$,
   $\{r' \mid$ if $j + 1 = i$ then $r'\colon \langle p, i, h\rangle \rightarrow \langle q'', i, h + m - 1\rangle$ else $r'\colon \langle p, j + 1,$
   $h\rangle \rightarrow \langle q', j + 1, h + m - 1\rangle\ \}$

$| 1 \leq j < i, i \leq h \leq h_{max}\}$

$\cup$

(ii) $\{\langle p,i,h\rangle \rightarrow \langle q'',i,h+m-1\rangle,$
$\{r' \mid$ if $i=h$, then $r'\colon \langle q'',i,h+m-1\rangle \rightarrow y_0\langle q',i,h+m-1\rangle y_1\langle q',i+1,h+m-1\rangle y_2 \ldots y_{m-1}\langle q',i+m-1,h+m-1\rangle y_m$ else $r'\colon \langle p,i+1,h\rangle \rightarrow \langle q',i+1+m-1,h+m-1\rangle\}$
$| i \leq h \leq h_{max}\}$

$\cup$

(iii) $\{\langle p,j,h\rangle \rightarrow \langle q',j+m-1,h+m-1\rangle,$
$\{r' \mid$ if $j=h$, then $r'\colon \langle q'',i,h+m-1\rangle \rightarrow y_0\langle q',i,h+m-1\rangle y_1\langle q',i+1,h+m-1\rangle y_2 \ldots y_{m-1}\langle q',i+m-1,h+m-1\rangle y_m$ else $r'\colon \langle p,j+1,h\rangle \rightarrow \langle q',j+1+m-1,h+m-1\rangle\}$
$| i < j \leq h, i \leq h \leq h_{max}\}$

$\cup$

(iv) $\{\langle q'',i,h+m-1\rangle \rightarrow y_0\langle q',i,h+m-1\rangle y_1\langle q',i+1,h+m-1\rangle y_2 \ldots y_{m-1}\langle q',i+m-1,h+m-1\rangle y_m,$
$\{r' \mid r'\colon \langle q',1,h+m-1\rangle \rightarrow \langle q,1,h+m-1\rangle\}$
$| i \leq h \leq h_{max}\}$

$\cup$

(v) $\{\langle q',j,h+m-1\rangle \rightarrow \langle q,j,h+m-1\rangle,$
$\{r' \mid$ if $j < h+m-1$, then $r'\colon \langle q',j+1,h+m-1\rangle \rightarrow \langle q,j+1,h+m-1\rangle$ else $r'\colon \langle \tilde{p},1,h+m-1\rangle \rightarrow \langle \tilde{q}',1,h+m-1+\tilde{m}-1\rangle$, where $\tilde{p}_{\tilde{i}}\# \rightarrow \tilde{q}\tilde{y}_0\#\tilde{y}_1 \ldots \tilde{y}_{\tilde{m}-1}\#\tilde{y}_{\tilde{m}} \in R$, $\tilde{y}_0,\tilde{y}_1,\ldots,\tilde{y}_{\tilde{m}} \in T^*$, if $\tilde{i}=1$, then $\tilde{q}' := \tilde{q}''\}$
$| 1 \leq j \leq h+m-1, i \leq h \leq h_{max}\}$.

Claim 3. If $\langle\sigma\rangle\#_d \Rightarrow^c \langle\vartheta\rangle y_0\#y_1 \ldots y_{n-1}\#y_n$ in H, then $S \Rightarrow^* y_0A_1y_1 \ldots y_{n-1}A_ny_n$ in G for some $c \geq 0$.

Basis: Let $c=0$. For $\langle\sigma\rangle\#_d \Rightarrow^0 \langle\sigma\rangle\#$ in H there exists $S \Rightarrow^0 S$ in G.

Induction Hypothesis: Suppose Claim 3 holds for all derivations of length c or less for some $c \geq 0$.

Induction Step: Consider $\langle\sigma\rangle\#_d \Rightarrow^c \langle\vartheta\rangle y_0\#y_1 \ldots y_{h-1}\#y_h\ [r_1r_2 \ldots r_c]$ in H, $r_t \in R$, $1 \leq t \leq c$ and $r_{c+1} \in R : \langle\vartheta\rangle_i\# \rightarrow \langle\omega\rangle x_0\#x_1 \ldots x_{m-1}\#x_m$, $x_0,\ldots,x_m \in T^*$ so that $\langle\vartheta\rangle y_0\# \ldots \#y_h\,{}_d \Rightarrow \langle\omega\rangle y_0\#y_1\# \ldots \#y_{i-1}x_0\#x_1\# \ldots \#x_my_i\#y_{i+1}\# \ldots \#y_h\ [r_{c+1}]$. Based on Claim 3 there exists also a derivation $D_{1*} : y_0A_1 \ldots A_hy_h \Rightarrow^* y_0A_1y_1 \ldots y_{i-1}x_0B_1x_1 \ldots B_mx_my_iA_{i+1} \ldots A_hy_h$ in G. It is shown such a derivation exists based on the construction part of the proof.
Let us have a form $y_0A_1y_1 \ldots A_hy_h$. Rename nonterminals A_t to $\langle\vartheta,t,t\rangle$ for $1 \leq t \leq h$ and get a base form $y_0\langle\vartheta,1,h\rangle y_1 \ldots y_{h-1}\langle\vartheta,h,h\rangle y_h$ which starts the simulation of the D_{1*} derivation. This simulation must come out of continuous application of construction's 4. item.

(4i) $\forall j : 1 \leq j < i$ apply $\langle p,j,h\rangle \rightarrow \langle q',j,h+m-1\rangle$:
$F_1 = y_0\langle\vartheta,1,h\rangle y_1 \ldots y_{h-1}\langle\vartheta,h,h\rangle y_h \Rightarrow y_0\langle\omega',1,h+m-1\rangle y_1\langle\vartheta,2,h\rangle y_2 \ldots y_{h-1}$
$\langle\vartheta,h,h\rangle y_h \Rightarrow^{i-2} y_0\langle\omega',1,h+m-1\rangle y_1 \ldots y_{i-2}\langle\omega',i-1,h+m-1\rangle y_{i-1}\langle\vartheta,i,h\rangle$
$y_i \ldots y_{h-1}\langle\vartheta,h,h\rangle y_h = F_2$

(4ii) apply $\langle p,i,h\rangle \rightarrow \langle q'',i,h+m-1\rangle$:
$F_2 \Rightarrow y_0\langle\omega',1,h+m-1\rangle y_1 \ldots y_{i-2}\langle\omega,i,h+m-1\rangle y_{i-1}y_i \ldots y_{h-1}\langle\vartheta,h,h\rangle = F_3$.
If $i=h$, then $F_4 := F_3$ and continue with [4iv] otherwise with [4iii].

(4iii) $\forall j : i < j \leq h$ apply $\langle p,j,h\rangle \rightarrow \langle q',j+m-1,h+m-1\rangle$:
$F_3 \Rightarrow y_0\langle\omega',1,h+m-1\rangle y_1 \ldots y_{i-2}\langle\omega',i-1,h+m-1\rangle y_{i-1}\langle\omega'',i,h+m-1\rangle y_i$

$\langle\omega', i+m, h+m-1\rangle y_{i+1}\langle\vartheta, i+2, h\rangle y_{i+2}\ldots y_{h-1}\langle\vartheta, h, h\rangle y_h \Rightarrow^{h-i-1} y_0\langle\omega', 1, h+m-1\rangle$
$y_1\ldots y_{i-1}\langle\omega'', i, h+m-1\rangle y_{i+1}\ldots y_{h-1}\langle\omega', h+m-1, h+m-1\rangle y_h = F_4$

(4iv) apply $\langle q'', i, h+m-1\rangle \rightarrow y_0\langle q', i, h+m-1\rangle y_1\ldots y_{m-1}\langle q', i+m-1, h+m-1\rangle y_m$:
$F_4 \Rightarrow y_0\langle\omega', 1, h+m-1\rangle y_1\ldots y_{i-1}x_0\ldots\langle\omega', i, h+m-1\rangle x_1\ldots x_{m-1}\langle\omega', i+m-1, h+m-1\rangle x_m y_i\ldots y_{h-1}\langle\omega', h+m-1, h+m-1\rangle y_h = F_5$

(4v) $\forall j : 1 \leq j \leq h+m-1$ apply $\langle q', j, m+m-1\rangle \rightarrow \langle q, j, h+m-1\rangle$:
$F_5 \Rightarrow^{h+m-1} y_0\langle\omega, 1, h+m-1\rangle y_1\ldots y_{i-1}x_0\langle\omega, i, h+m-1\rangle x_1\ldots x_{m-1}\langle\omega, i+m-1, h+m-1\rangle x_m y_i\ldots y_{h-1}\langle\omega, h+m-1, h+m-1\rangle y_h = F_6$ (Final form)

Rename all nonterminals of the form $\langle\omega, t, h+m-1\rangle$ in F_6 to A_t, where $1 \leq t < i$, $\langle\omega, t, h+m-1\rangle$ to B_{t-i+1}, where $i \leq t \leq i+m$, $\langle\omega, t, h+m-1\rangle$ to A_{t-m+1}, where $i+m < t \leq h+m-1$. We have obtained $rhs(D_{1*})$.

Claim 4. If $\langle\sigma\rangle\#\ {}_d\Rightarrow^z\ \langle\rangle x$ in H, then $S \Rightarrow^* x$ for some $z \geq 0$.
This Claim follows from Claim 3 for $n = 0$. □

4.2 Nekonečná hierarchie řetězce rozdělujících systémů s konečným indexem

Theorem 1. Infinite hierarchy $\mathcal{L}_k(SPS,\ {}_d\Rightarrow) \subset \mathcal{L}_{k+1}(SPS,\ {}_d\Rightarrow)$ holds for every $k \geq 1$.
Proof. $\mathcal{L}_k(P, CF) = \mathcal{L}_k(SPS,\ {}_d\Rightarrow)$ follows from Lemma 1 and 2. Then, Theorem 1 follows from $\mathcal{L}_k(P, CF) = \mathcal{L}_k(SPS,\ {}_d\Rightarrow)$ and theorem $\mathcal{L}_k(P, CF) \subset \mathcal{L}_{k+1}(P, CF)$, for every $k \geq 1$ which is an analogy to Theorem 3.1.7: $\mathcal{L}_k(M, CF) \subset \mathcal{L}_{k+1}(M, CF)$ in [1], page 161. ■

5 Syntaktická analýza

5.1 Řízené gramatiky

Programované gramatiky a obecně - řízené gramatiky vůbec - vykazují podobné rysy nedeterminismu v průběhu derivace jako gramatiky bezkontextové. Jednak může být použito více pravidel v daném kroku a navíc se levá strana vybraného pravidla může ve větné formě vyskytovat vícekrát. V bezkontextových gramatikách je rozhodování mezi pravidly v každém kroku, použita mohou být totiž všechna pravidla, jejichž levá strana se vyskytuje ve větné formě. Řízené gramatiky se snaží minimalizovat toto množství pravidel a tím i nutnost častého výběru.

Při praktickém použití přepisovacích systémů v syntaktické analýze je výhodné omezit možnosti přepisování na nejlevější, resp. nepravější derivace. Vede to ke zjednodušení a zefektivnění algoritmů. Některé typy řízených gramatik disponují možností tzv. nejlevějšího omezení. Tento pojem byl zaveden pro maticové gramatiky. Nejlevější derivace vždy vybírá pro přepsání nejlevější neterminální symbol, na kterou jde matici použít.

Stávající příklady pro popis programovacích jazyků pomocí programovaných gramatik vycházejí obvykle z použití obou, pole úspěchu i pole neúspěchu. V tomto případě ovšem síla programovaných gramatik stoupne až na sílu rekurzivně spočetných jazyků. Na druhou stranu, pokud by v rámci těchto příkladů postačoval konečný index dané gramatiky, mohli bychom pro ně sestrojit i odpovídající řetězce rozdělující systémy, je však otázka, zda-li toto snížení generativní síly bude dostačující pro všechny použité jazykové konstrukce.

5.2 Řetězce rozdělující systémy

Řetězce rozdělující systémy vykazují jeden specifický rys, který je významně odlišuje od klasických gramatik. V klasických - např. bezkontextových gramatikách - figurují neterminální symboly ve dvojím významu. Za prvé určují pozici ve větné formě pro možný přepis v následující derivaci, zároveň však vyčleňují množinu pravidel (ze všech dostupných pravidel), která mohou být aplikována. Tyto dva významy slučuje do jednoho konkrétního neterminálního symbolu. Mechanismus výběru místa přepisování je v řetězce rozdělujících systémech odlišný. Je zde ekvivalent neterminálu (značka) ve významu určení pozice pozdějšího přepisu, avšak druhou informaci - tj. vyčlenění množiny aplikovatelných pravidel už zastává kombinace informací: pozice i-té značky (z pohledu i-té značky, která si svou pozici ovšem sama nepamatuje) a specifikace i-té značky na levé straně příslušných pravidel.

Na syntaktickou analýzu má samozřejmě tato skutečnost obrovský vliv. Buď navrhneme řetězce rozdělující systém pro syntaktickou analýzu tak, aby zpracovával příslušné věty jazyka vždy přísně zleva doprava, nebo nalezneme mechanismus, který nám v rámci existujících možností řetězce rozdělujících systémů dokáže kódovat vlastnosti řízených systémů s konečným indexem.

Druhý případ vnímejme spíše jako teoretický pokus, i kdyby se nám podobný mechanismus podařilo najít, bude v praxi nepoužitelný a bude nemyslitelné vystavět na něm prakticky syntaktickou analýzu. Ke každé značce ve větné formě (resp.v pravidlech) bychom museli připojit informaci o na ni aplikovatelných pravidlech.

Syntaktickou analýzu je možné založit už na bezkontextových jazycích. Vzhledem k tomu, že bezkontextové jazyky jsou neporovnatelné s jazyky generovanými řetězce rozdělujícími systémy s konečným indexem, musíme nejdříve ověřit, zda-li jsme jimi schopni popsat všechny požadované konstrukce vyskytující se v popisovaném jazyce, a poté sestrojit odpovídající řetězce rozdělující systém. Je otázkou, jestli tento přístup bude mít oproti použití bezkontextových jazyků nějaké výhody: generativní síla, snadnější návrh..

6 Závěr

Tato práce vychází z již existujících materiálů o řetězce rozdělujících systémech, zahrnuje však i části připravovaného článku o zařazení řetězce rozdělujících systémů s konečným indexem do kontextu řízených gramatik s konečným indexem. Cílem této práce bylo jednak podat formální zápis důkazu ekvivalence programovaných gramatik a řetězce rozdělujících systémů se stejným indexem, ale také zpřístupnit hlavní myšlenky důkazů v rámci přiložených komentářů. Bylo dokázáno, že jazyky popsané řetězce rozdělujícími systémy s indexem k pro všechna $k \geq 1$ tvoří nekonečnou hierarchii jazyků.

V závěru je diskutováno použití řetězce rozdělujících systémů na poli syntaktické analýzy a vyjádřena potřeba podrobnějšího zkoumání jejich popisných možností zhlediska vyjádření jednotlivých typů jazykových konstrukcí v programovacích jazycích.

Bibliography

1. Dassow, J., Păun, G.: Regulated Rewriting in Formal Language Theory, Springer, New York, 1989.
2. Meduna, A.: Automata and Languages: Theory and Applications, Springer, London, 2000.
3. Křivka, Z.: String-partitioning systems – essay in course Modern Theoretical Computer Science [in czech], FIT Brno University of Technology, 2004.
4. Rozendberg, G., Salomaa, A. (eds.): *Handbook of Formal Languages*, Volumes 1-3, Springer, Berlin, 1997.
5. Kot, M.: *Řízené gramatiky*, diplomová práce, VŠB, Fakulta elektrotechniky a informatiky, 2002