pokladem, že gramatika $\mathscr{G}$ je redukovaná. M proto obsahuje jedinou trojici $([A], \alpha, [x])$, jejíž třetí složka je $[x]$. Proto $[M_x] = \{([A], \alpha x, c)\}$, kde c je jednoznačně určeno podle lemmatu 5.3.4. $[M_x]$ je jednobodová, a proto má souvislou charakteristiku.

Důsledek 5.3.9. *Každý stav $I(\gamma)$ položkového automatu pro gramatiku $\mathscr{G}$ má souvislou charakteristiku.*

Důkaz. Vyplývá okamžitě z lemmat 5.3.7, 5.2.13, 5.3.6 a 5.3.8.

Z předchozího tvrzení už se snadno dokáže, že $\mathscr{G}$ je LR(0) gramatika.

Důkaz věty 5.3.1. Z definice množiny souvislé charakteristiky vyplývá, že taková množina může obsahovat nejvýše jednu trojici, jejíž třetí složka je e nebo Σ. Každá úplná položka $A \to \alpha.$ má charakteristiku $([A], \alpha, e)$. Každá položka $A \to \alpha.\beta$, v níž je napravo od tečky terminál, má charakteristiku $([A], \alpha, \Sigma)$. Proto každá množina souvislé charakteristiky může zahrnovat nejvýše jednu úplnou položku a v tom případě žádnou položku s terminálem bezprostředně napravo od tečky.

Každá množina $I(\gamma)$ je podle tvrzení 5.3.9 souvislé charakteristiky, a proto splňuje podmínku 2 definice LR(0) gramatiky (viz 5.2.19). Podmínka 1 je splněna také, neboť počáteční neterminál S se v žádném pravidle nevyskytuje na pravé straně.

Z vět 5.2.26 a 5.3.1 dostáváme následující výsledek.

Věta 5.3.10. *Jazyk lze popsat* LR(0) *gramatikou, právě když jej lze rozpoznat deterministickým zásobníkovým automatem prázdným zásobníkem.*

5.4. LR(k) gramatiky

V čl. 5.3 jsme zavedli položkový automat, jakožto zdroj informace použitelné k deterministické analýze LR(0) gramatik. Stavy tohoto automatu jsou množiny položek.

Nyní zavedeme obecnější pojem k-položky, který umožní příslušnou informaci získávat pro analýzu širší třídy gramatik.

Definice 5.4.1. Nechť $\mathscr{G} = (\Pi, \Sigma, S, P)$ je bezkontextová gramatika a $k \geqq 1$ přirozené číslo. *k-položkou* gramatiky $\mathscr{G}$ nazveme

každou dvojici $(A \to \alpha.\beta, L)$, kde $A \to \alpha.\beta$ je položka $\mathscr{G}$ a $L \subseteq \Sigma^{*k}$. Symbolem Σ^{*k} označujeme množinu všech slov v abecedě Σ, jejichž délka nepřesahuje k. V dalším textu už budeme odkaz na gramatiku $\mathscr{G}$ vynechávat.

Definice 5.4.2. Řekneme, že k-položka $(A \to \alpha.\beta, L)$ je *platná k-položka* pro řetěz $\omega \in (\Sigma \cup \Pi)^*$, jestliže existuje pravá větná forma $\eta A u$ taková, že $\eta\alpha = \omega$ a $k(u) \in L$.

Poznámka 5.4.3. Jestliže $(A \to \alpha.\beta, L)$ je platná k-položka pro řetěz ω, potom je zřejmě $A \to \alpha.\beta$ platná položka pro ω.

Úmluva 5.4.4. Symbolem $I_k(\gamma)$ budeme označovat množinu všech platných k-položek pro řetěz γ.

Definice 5.4.5. Nechť $\mathscr{K}'$ je množina všech množin k-položek (pro danou gramatiku). Definujme funkci $\delta' : \mathscr{K}' \times (\Sigma \cup \Pi) \to \mathscr{K}'$ takto: pro každé $K \in \mathscr{K}'$ a $x \in (\Sigma \cup \Pi)$ je $\delta'(\mathscr{K}, x)$ nejmenší množina M taková, že

1. $M \supseteq \{(Y \to \alpha x.\beta, L);\ (Y \to \alpha.x\beta, L) \in K\}$;
2. jestliže $(A \to \alpha.B\beta, L) \in M$, potom pro každé pravidlo $B \to \vartheta$ gramatiky je $(B \to .\vartheta, k(\beta L)) \in M$.

Definice 5.4.6. Definujme množinu $K_0 \in \mathscr{K}'$, kde $\mathscr{K}'$ má týž význam jako v předchozí definici, jakožto nejmenší množinu M takovou, že

1. $M \supseteq \{(S \to .\alpha, \{e\});\ S \to \alpha \in P\}$;
2. jestliže $(A \to .B\beta, L) \in M$, potom také pro každé pravidlo $B \to \vartheta \in P$ je $(B \to .\vartheta, k(\beta L)) \in M$.

Symbolem $\mathscr{K}$ označme množinu prvků množiny $\mathscr{K}'$ dosažitelných z K_0 (přechodovou funkcí δ') a symbolem δ označme restrikci δ' na $\mathscr{K} \times (\Sigma \cup \Pi)$.

Konečný automat

$$\mathscr{A} = (\mathscr{K}, \Sigma \cup \Pi, \delta, K_0, F),$$

kde $F = \mathscr{K} - \{\emptyset\}$, nazveme *$k$-položkovým automatem pro danou gramatiku.*

Definice 5.4.7. Nechť $k > 0$. Bezkontextová gramatika

$$\mathscr{G} = (\Pi, \Sigma, S, P)$$

je LR(k) *gramatika*, jestliže platí tyto tři podmínky:

1. Počáteční symbol se nevyskytuje na pravé straně žádného pravidla;

2. jestliže $(A \to \alpha., L_1)$, $(B \to \beta., L_2)$ jsou dvě k-položky (úplné k-položky) vyskytující se současně v nějakém stavu $K \in \mathscr{K}$ a takové, že $L_1 \cap L_2 \neq \emptyset$, potom pravidlo $A \to \alpha$ je totožné s pravidlem $B \to \beta$;

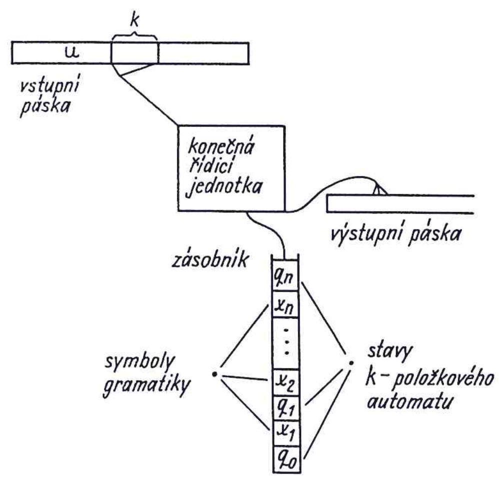
3. jestliže se v některém stavu $K \in \mathscr{K}$ současně s úplnou k-položkou $(A \to \alpha., L_1)$ vyskytne k-položka tvaru $(B \to \beta.\gamma, L_2)$, v níž se bezprostředně napravo od tečky vyskytuje terminál, potom $L_1 \cap k(\gamma L_2) = \emptyset$.

Poznámka 5.4.8. Z definic 5.2.17 a 5.4.6 vyplývá, že pro každý řetěz γ je

$$\{A \to \alpha.\beta;\ (A \to \alpha.\beta, L) \in \delta(K_0, \gamma) \text{ pro nějaké } L\} \subseteq I(\gamma).$$

Z tohoto zjištění a z definice 5.4.7 pak plyne, že každá LR(0) gramatika je LR(k) gramatikou pro libovolné $k \geqq 1$.

Analogicky lze dokázat, že každá LR(k) gramatika je LR($k + 1$) pro každé $k \geqq 0$.



Obr. 43

Jazyk generovaný libovolnou LR(k) gramatikou

$$\mathscr{G} = (\Pi, \Sigma, S, P)$$

je možné analyzovat typem deterministického analyzátoru, který je schematicky zachycen na obr. 43.

Takový analyzátor je modifikací deterministického automatu s výstupní páskou. Změna se týká pouze vstupní hlavy, která v tomto případě snímá zároveň obsah k sousedních políček. Po zpracování libovolného počátečního úseku vstupního slova má proto LR(k) *analyzátor* k dispozici informaci i o následující k-tici vstupních symbolů. Zásobník je obsluhován obdobným způsobem, s jakým jsme se setkali už v čl. 5.3 u deterministické analýzy LR(0) jazyků. Díky tomu je v každém okamžiku výpočtu obsah zásobníku tvaru

$$q_0 x_1 q_1 x_2 q_2 \dots q_{n-1} x_n q_n$$

(pravý konec opět odpovídá vrcholu zásobníku), kde $q_0, \dots, q_n$ jsou stavy k-položkového automatu pro $\mathscr{G}$, $x_1, \dots, x_n \in \Sigma \cup \Pi$, $n \geqq 0$, q_0 je počáteční stav a

$$q_i = \delta(q_0, x_1 \dots x_i)$$

pro všechna i, $1 \leqq i \leqq n$.

Způsob, jakým se obsah zásobníku aktualizuje při přenosu dalšího vstupního symbolu do zásobníku i při redukci obsahu podle některého pravidla gramatiky zůstává naprosto stejný jako u LR(0) analýzy.

Změna nastává pouze u postupu, kterým se v každém kroku výpočtu rozhoduje, zda dojde k přesunu do zásobníku nebo k redukci a ve druhém případě pak ještě podle kterého pravidla se redukce provádí.

U LR(0) gramatik se příslušné rozhodování provádělo na základě položky nacházející se na vrcholu zásobníku. U LR(k) analyzátoru se další postup určí podle těchto dvou údajů:

1. množiny q k-položek nacházející se na vrcholu zásobníku;
2. následující k-tice u vstupních symbolů.

Z nich se příští krok určí takto:

a) Jestliže q obsahuje k-položku $(A \to \alpha., L)$ takovou, že $u \in L$

a A není počáteční symbol, provede se redukce podle pravidla $A \to \alpha$. [Podrobněji: ze zásobníku se odebere vrchních $2 . |\alpha|$ symbolů, tj. symboly tvořící α a stavy umístěné nad každým z nich. Tím se na vrcholu zásobníku objeví jistý stav q'. Nad něj se uloží dvojice symbolů Aq'', kde $q'' = \delta(q', A)$. Na výstupu se vytiskne číslo pravidla $A \to \alpha$.]

Jestliže A je počáteční symbol, ukončí se po provedení redukce výpočet vyprázdněním zásobníku a přijetím slova.

b) Jestliže q obsahuje k-položku $(A \to \alpha . \beta, L)$ takovou, že prvním symbolem β je terminál a $u \in k(\beta L)$, potom se provede přesun do zásobníku. [Podrobněji: do zásobníku se přesune další vstupní symbol b – který je nutně totožný s prvním symbolem β – a nad něj se uloží symbol $\delta(q, b)$.]

c) Jestliže nenastane ani možnost a), ani b), ukončí se výpočet zamítnutím zkoumaného slova.

Vlastnosti LR(k) gramatiky požadované v definici 5.4.7 zaručují, že ze všech eventualit uvedených sub a) až c) nastane právě jedna.

Příklad 5.4.9. Přesvědčíme se, že gramatika $\mathscr{G}$:

1. $S \to AB$,
2. $S \to A$,
3. $A \to aAb$,
4. $A \to e$,
5. $B \to bB$,
6. $B \to c$

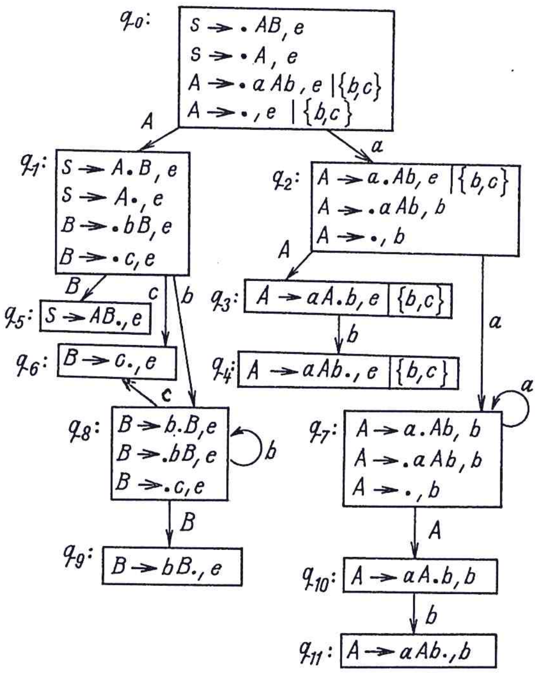
je LR(1) gramatika a sestrojíme k ní LR(1) analyzátor.

Diagram 1-položkového automatu pro $\mathscr{G}$ je sestrojen na obr. 44. Na diagramu jsme opět pro přehlednost vypustili stav odpovídající prázdné množině 1-položek (zde by to byl stav q_{12}) a všechny přechody do něj vedoucí. V diagramu je použito vžitého zkratkového značení. Jednoprvkový jazyk $\{x\}$ značíme pouze x; množinu položek

$$(A \to \alpha . \beta, L_1), \ldots, (A \to \alpha . \beta, L_n)$$

zkráceně zapisujeme

$$A \to \alpha . \beta, L_1 \mid L_2 \mid \ldots \mid L_n .$$



Obr. 44

Gramatika $\mathscr{G}$ je LR(1), neboť počáteční symbol se nevyskytuje na pravé straně žádného pravidla a 1-položkový automat na obr. 44 také splňuje podmínky z definice 5.4.7. Tak např. ve stavu q_0 jsou obsaženy dvě úplné položky

$$p_1 = (A \to ., e) \text{ a } p_2 = (A \to ., \{b, c\})$$

a dvě položky s terminálem bezprostředně napravo od tečky:

$$p_3 = (A \to .aAb, e) \text{ a } p_4 = (A \to .aAb, \{b, c\}) .$$

Ke konfliktu mezi p_1 a p_2 (viz podmínku 2 z definice 5.4.7) nedochází, neboť odpovídají témuž pravidlu a navíc

$$\{e\} \cap \{b, c\} = \emptyset .$$

Podobně nedochází ke konfliktu mezi p_1 a p_3 nebo p_4 (podmínka 3), protože

$$\{e\} \cap 1(aAb\{e\}) = \{e\} \cap 1(aAb\{b, c\}) = \emptyset .$$

Ke konfliktu mezi p_2 a p_3 nebo p_4 nedochází, protože

$$\{b, c\} \cap 1(aAb\{e\}) = \{b, c\} \cap 1(aAb\{b, c\}) = \emptyset .$$

Na základě 1-položkového automatu z obr. 44 lze tedy sestrojit LR(1) analyzátor pro gramatiku $\mathscr{G}$. Akce tohoto analyzátoru v závislosti na vrchním symbolu v zásobníku (tj. stavu 1-položkového

	Vstupní symbol			
	a	b	c	e
q_0	**přesun** a	$A \to e$; 4	$A \to e$; 4	$A \to e$; 4
q_1		**přesun** b	**přesun** c	**přijmout**; 2
q_2	**přesun** a	$A \to e$; 4		
q_3		**přesun** b		
q_4		$A \to aAb$; 3	$A \to aAb$; 3	$A \to aAb$; 3
q_5				**přijmout**; 1
q_6				$B \to c$; 6
q_7	**přesun** a	$A \to e$; 4		
q_8		**přesun** b	**přesun** c	
q_9				$B \to bB$; 5
q_{10}		**přesun** b		
q_{11}		$A \to aAb$; 3		
q_{12}				

Tab. 15

automatu) a následujícím vstupním symbolu (nebo e, což odpovídá tomu, že slovo bylo dočteno do konce) popíšeme tab. 15.

Akce označovaná v tabulce „$X \to \eta$; i" znamená: zredukovat podle pravidla $X \to \eta$ (včetně příslušné aktualizace zásobníku) a vytisknout i. Vstupní hlava se neposouvá.

Akce označovaná „**přijmout**; i" znamená: vyprázdnit zbytek zásobníku a vytisknout i. Výpočet končí přijetím slova.

Akce označovaná „**přesun** x" znamená: uložit symbol x do zásobníku (a přidat nad něj příslušný symbol stavu položkového automatu) a posunout vstupní hlavu.

Prázdná políčka v tabulce označují situace, ve kterých analyzátor končí a vydá hlášení, že analyzované slovo nepatří do $L(\mathscr{G})$. Speciálně taková situace nastává vždy, když se na zásobníku objeví symbol q_{12} odpovídající prázdné množině položek.

Příklad 5.4.10. Nad slovem *aabbc* pracuje analyzátor sestrojený v př. 5.4.9 takto (první člen trojice označuje vždy zbytek vstupního slova, druhý člen obsah zásobníku a třetí obsah výstupní pásky):

$$(aabbc; q_0; e) \vdash (abbc; q_0aq_2; e) \vdash$$
$$\vdash (bbc; q_0aq_2aq_7; e) \vdash (bbc; q_0aq_2aq_7Aq_{10}; 4) \vdash$$
$$\vdash (bc; q_0aq_2aq_7Aq_{10}bq_{11}; 4) \vdash (bc; q_0aq_2Aq_3; 43) \vdash$$
$$\vdash (c; q_0aq_2Aq_3bq_4; 43) \vdash (c; q_0Aq_1; 433) \vdash$$
$$\vdash (e; q_0Aq_1cq_6; 433) \vdash (e; q_0Aq_1Bq_5; 4336) \vdash$$
$$\vdash \text{slovo je přijato, jeho pravý rozbor je } 43361 .$$

5.5. LR jazyky a deterministické jazyky

Definice 5.5.1. Jazyk nazveme LR(k) *jazykem*, jestliže existuje LR(k) gramatika, která jej generuje. Řekneme, že jazyk je LR *jazykem*, jestliže je LR(k) jazykem pro nějaké přirozené číslo $k \geqq 0$.

Cílem tohoto článku je dokázat, že libovolný jazyk je LR, právě když je deterministický. Jedna část tvrzení plyne z toho, že LR(k) analyzátory lze simulovat deterministickým zásobníkovým automatem.

Věta 5.5.2. *Každý* LR *jazyk je deterministický.*

D ů k a z. V článku 5.2 jsme ukázali, jak převést LR(0) analyzátor na deterministický zásobníkový automat. Jednalo se o způsob obsluhování zásobníku (pozn. 5.2.25) a tentýž způsob lze využít při přechodu od LR(k) analyzátoru k deterministickému zásobníkovému automatu pro libovolné $k \geqq 0$. V případě $k \geqq 1$ vyvstává ještě problém, jak nahradit možnost prohlížení k symbolů dopředu na vstupní pásce, kterou má LR(k) analyzátor. V možnostech deterministického automatu je uchovávat informaci o k-tici symbolů v konečné paměti řídicí jednotky. Deterministický automat tedy může např. provádět manipulaci se zásobníkem se zpožděním až po přečtení potřebných k symbolů. Zde ovšem vzniká další potíž. Kdyby automat pouze udržoval konstantní předstih hlavy, mohlo by se stát, že hlava opustí pravý konec slova dříve, než má automat možnost dokončit simulaci výpočtu. Tomu lze poměrně snadno odpomoci, pokud je pravý konec vstupní pásky označen speciálním koncovým znakem. Na tomto znaku se vstupní hlava zastaví.

Od deterministického automatu s koncovým znakem lze pak přejít k ekvivalentnímu deterministickému zásobníkovému automatu bez koncového znaku, protože podle tabulky 8 jsou deterministické jazyky uzavřeny vůči pravému kvocientu s regulárním jazykem.

Ukážeme, že platí i věta obrácená k větě 5.5.2.

Věta 5.5.3. *Každý deterministický jazyk je* LR(1) *jazyk.*

D ů k a z věty je bezprostředním důsledkem následujících tří lemmat.

Lemma 5.5.4. *Nechť L je libovolný bezkontextový jazyk, $\#$ přidaný symbol a $\mathscr{G}$ redukovaná bezkontextová gramatika generující jazyk $L\#$. Jestliže některý stav 1-položkového automatu pro $\mathscr{G}$ obsahuje 1-položku tvaru $(A \to \alpha\,.\,\#\,\beta, \hat{L})$, potom $\hat{L} = \{e\}$ a $1(\beta) = e$.*

D ů k a z. Indukcí se snadno ověří, že kdyby existovalo neprázdné slovo $u \in \hat{L}$, nebo $1(\beta) \neq e$, existovalo by odvození

$$S \Rightarrow^* \delta A u v \Rightarrow^* \delta\alpha\,\#\,\beta u v \Rightarrow^* w_1\,\#\,w_2$$

pro nějaké $w_2 \neq e$, což je spor s předpokladem, že $w_1\,\#\,w_2 \notin L\#$.

Lemma 5.5.5. *Nechť $L\#$ je* LR(0) *jazyk, přičemž symbol $\#$ se nevyskytuje v žádném slově jazyka L. Potom L je* LR(1) *jazyk.*

D ů k a z. Nechť $\mathscr{G}$ je LR(0) gramatika taková, že $L(\mathscr{G}) = L\#$. Sestrojme 1-položkový automat $\mathscr{A}$ pro $\mathscr{G}$. Jakmile některý stav q automatu $\mathscr{A}$ obsahuje 1-položku tvaru $(A \to \alpha\,.\,\#\,\beta, \hat{L})$, je podle 5.5.4 $\hat{L} = \{e\}$. Navíc platí, že q neobsahuje žádnou úplnou 1-položku. Jinak by odpovídající stav položkového automatu obsahoval jednak $A \to \alpha\,.\,\#\,\beta$, jednak úplnou položku, a to by bylo ve sporu s předpokladem, že $\mathscr{G}$ je LR(0) gramatika. Podobně q neobsahuje žádnou další 1-položku s terminálem bezprostředně za tečkou.

Sestrojme nyní gramatiku $\mathscr{G}'$, která se od $\mathscr{G}$ bude lišit pouze tím, že $\#$ bude v $\mathscr{G}'$ patřit mezi neterminály a oproti $\mathscr{G}$ bude mít navíc pravidlo $\# \to e$.

Okamžitě je zřejmé, že $L(\mathscr{G}') = L$.

Z konstrukce 1-položkového automatu plyne, že 1-položkový automat pro $\mathscr{G}'$ se od 1-položkového automatu pro $\mathscr{G}$ liší pouze ve stavech, které u gramatiky $\mathscr{G}$ obsahovaly 1-položku typu $(A \to \alpha\,.\,\#\,\beta, e)$. Odpovídající stav u $\mathscr{G}'$ vždy navíc obsahuje 1-položku $(\# \to .\,, e)$. Ostatní stavy se shodují. Z toho je vidět, že i 1-položkový automat pro $\mathscr{G}'$ splňuje podmínky z definice LR(1) gramatiky. Stavy, ve kterých je nyní navíc 1-položka $(\# \to .\,, e)$, už u gramatiky $\mathscr{G}$ neobsahovaly jinou úplnou 1-položku. Pokud takový stav obsahuje nějakou 1-položku $(X \to \gamma\,.\,x\delta, L)$ s terminálem x, potom kolize s 1-položkou $(\# \to .\,, e)$ opět nenastává, protože

$$1(x\delta\,.\,\hat{L}) = \{x\} \text{ a } \{x\} \cap \{e\} = \emptyset\,.$$

[Samozřejmě nenastává ani kolize s $(A \to \alpha\,.\,\#\,\beta, e)$, protože $\#$ je v $\mathscr{G}'$ neterminálem.]

Lemma 5.5.6. *Jestliže L je deterministický jazyk a $\#$ nově přidaný symbol, potom $L\#$ je* LR(0) *jazyk.*

D ů k a z. Jazyk $L\#$ je podle 4.4.11 rozpoznatelný deterministickým zásobníkovým automatem pomocí prázdného zásobníku. Proto je podle 5.3.1 LR(0) jazyk.

D ů k a z věty 5.5.3. Buď L libovolný deterministický jazyk a $\#$ nějaký nově přidaný symbol. Potom $L\#$ je podle 5.5.6 LR(0) jazyk, a tedy L je podle 5.5.5 LR(1) jazyk.

Věta 5.5.7. *Jazyk je deterministický, právě když je to* LR *jazyk.*

D ů k a z plyne bezprostředně z vět 5.5.2 a 5.5.3.

Důsledek 5.5.8. *Každý* LR *jazyk je* LR(1) *jazyk.*

D ů k a z. Každý LR jazyk je deterministický podle 5.5.2, a tedy jej podle 5.5.5 generuje vhodná LR(1) gramatika.

Poznámka 5.5.9. Z 5.5.7 pochopitelně neplyne, že každá LR gramatika je LR(1). Naopak, lze ukázat, že pro každé $k > 0$ existuje LR(k) gramatika, která není LR($k - 1$).

5.6. LL(k) gramatiky

Zobecněním pojmu LL(1) gramatiky dostaneme pojem LL(k) gramatiky.

Definice 5.6.1. Budiž $k \geqq 1$ přirozené číslo. O bezkontextové gramatice $\mathscr{G} = (\Pi, \Sigma, S, P)$ říkáme, že je LL(k) *gramatika*, jestliže pro libovolná dvě pravidla $A \to \alpha$, $A \to \beta$ $(\alpha \neq \beta)$ gramatiky a libovolnou levou větnou formu $uA\eta$ $(u \in \Sigma^*)$ platí, že

$$k(\alpha\eta) \cap k(\beta\eta) = \emptyset\,.$$

Čtenář, který už si osvojil, jak souvisí levé derivace s analýzou shora, dokáže jistě odhadnout dosah této definice pro analýzu. Levá větná forma $uA\eta$ odpovídá situaci, kdy v zásobníku je uloženo slovo $A\eta$ (levý konec slova zde opět znamená vrchol zásobníku). Jestliže je splněna podmínka z definice LL(k) gramatiky, znamená to, že k-tice terminálních symbolů, která se postupně objeví na vrcholu zásobníku $\alpha\eta$ (tj. zásobníku po přepsání podle pravidla $A \to \alpha$), se v žádném případě nemůže objevit na vrcholu při zpracování zásobníku $\beta\eta$ a obráceně. Klíč k určení, které z pravidel $A \to \alpha$, $A \to \beta$ použít, tedy lze najít v následující k-tici vstupních symbolů.

Definice 5.6.2. Budiž $k \geqq 1$ přirozené číslo. *Silnou* LL(k) *gramatikou* nazveme každou bezkontextovou gramatiku

$$\mathscr{G} = (\Pi, \Sigma, S, P)$$

splňující tuto podmínku: Pro libovolná dvě pravidla $A \to \alpha$, $A \to \beta$ $(\alpha \neq \beta)$ gramatiky a libovolné levé větné formy $uA\eta$, $vA\vartheta$ $(u, v \in \Sigma^*)$ je

$$k(\alpha\eta) \cap k(\beta\vartheta) = \emptyset\,.$$

Poznámka 5.6.3. Než vyjasníme zdánlivý nesoulad mezi definicí LL(1) gramatiky z čl. 5.1 a definicemi 5.6.1, 5.6.2, uvědomme si, že požadavek kladený na silnou LL(k) gramatiku je silnější než požadavek kladený na LL(k) gramatiku. Proto je každá silná LL(k) gramatika také LL(k) gramatikou. Obráceně to platit nemusí (pro $k \geqq 2$), jak ukážeme.

Příklad 5.6.4. Přesvědčíme se, že gramatika

$$S \to 0A00 \mid 1A10\,,$$

$$A \to 1 \mid e$$

je LL(2), ale není to silná LL(2) gramatika. S-pravidla mají na začátku pravých stran odlišné terminály, takže stačí prozkoumat pouze A-pravidla. Existují pouze dvě levé formy obsahující A, a to $0A00$ a $1A10$. Ověřme pro obě podmínku z definice 5.6.1: $2(100) \cap 2(e00) = \emptyset$ a $2(110) \cap 2(e10) = \emptyset$, gramatika je tedy LL(2).

Na druhé straně $2(100) \cap 2(e10) = \{10\}$, což znamená, že gramatika není silná LL(2) gramatika.

Poznámka 5.6.5. Uchýlíme-li se opět k interpretaci definice 5.6.2 v termínech analýzy shora, je možné silné LL(k) gramatiky charakterizovat jako takové, u nichž přepsání symbolu A na vrcholu zásobníku řetězem α při nějakém výpočtu vede k postupnému vydání k-tice terminálních symbolů odlišných od k-tic, které se objeví po přepsání A na β i při libovolném jiném výpočtu. Díky tomu je návrh analyzátorů pro silné LL(k) gramatiky jednodušší než v obecném LL(k) případě.