Syntaktická analýza zdola nahor (1. časť)

Peter Kostolányi

29. apríla 2025

Syntaktickú analýzu (angl. tiež *parsing*) možno zhruba charakterizovať ako proces, ktorého hlavným cieľom je pre danú (obyčajne bezkontextovú) gramatiku G a slovo w zistiť, či $w \in L(G)$ – v prípade kladnej odpovede je žiadúcim výstupom syntaktickej analýzy aj nejaký strom odvodenia slova w v gramatike G; v prípade $w \notin L(G)$ zas nejaká informácia o príčine tejto skutočnosti. Algoritmy realizujúce túto úlohu nachádzajú uplatnenie pri tvorbe kompilátorov.

V nasledujúcom sa zameriame výlučne na rozhodovanie príslušnosti slova w do jazyka $L(G)$; všetky opísané algoritmy ale bude možné ľahko upraviť tak, aby súčasťou výstupu bol aj strom odvodenia resp. informácia o príčine jeho neexistencie.

Algoritmus CYK realizuje opísanú úlohu pre všeobecné bezkontextové gramatiky. Počas jeho vykonávania je ale nutné udržiavať v pamäti tabuľku o veľkosti kvadratickej od dĺžky slova w, ktorá môže byť v praxi veľmi veľká. Podobne môže predstavovať problém aj kubická časová zložitosť algoritmu CYK. Pre praktické účely je preto nutné (alebo v súčasnosti aspoň maximálne výhodné) mať k dispozícii algoritmus, ktorý syntaktickú analýzu realizuje s menšími pamäťovými a časovými nárokmi. Metódy syntaktickej analýzy zdola nahor, založené na schéme „posuň a redukuj“, sú v praxi najrozšírenejšími príkladmi takýchto efektívnych algoritmov. Tieto na rozdiel od algoritmu CYK nerealizujú syntaktickú analýzu pre všeobecné bezkontextové gramatiky, ale iba pre gramatiky s určitými vlastnosťami – to je ale pre praktické účely obvykle plne postačujúce.

1 Syntaktická analýza zdola nahor

Algoritmy syntaktickej analýzy zdola nahor pracujú na báze *spätnej* simulácie odvodenia daného slova w v danej gramatike G. Slovo w teda najprv (nejakým zatiaľ bližšie nešpecifikovaným spôsobom) upravia na slovo w' také, že $w' \Rightarrow w$, slovo w' ďalej upravia na slovo w'' také, že $w'' \Rightarrow w'$ a tak ďalej. V prípade, že algoritmus takýmto spôsobom zredukuje slovo w na počiatočný neterminál σ, platí $\sigma \Rightarrow^* w$, a teda $w \in L(G)$. Algoritmy syntaktickej analýzy zdola nahor teda postupujú od listov stromu odvodenia (ktorých zreťazenie je w) k jeho koreňu (počiatočnému neterminálu σ), čo odôvodňuje ich pomenovanie.

Väčšina algoritmov syntaktickej analýzy zdola nahor číta slovo w postupne *zľava doprava*, pričom sa konštruuje *pravé krajné* odvodenie slova w.

Označenie 1. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. Symbolom $\Rightarrow_{rm}$ označujeme reláciu *kroku pravého krajného odvodenia* v gramatike G. Pre $u, v \in (N \cup T)^*$ je teda $u \Rightarrow_{rm} v$ práve vtedy, keď pre nejaké $u_1, x \in (N \cup T)^*$, $u_2 \in T^*$ a $\xi \in N$ je $u = u_1 \xi u_2$, $v = u_1 x u_2$ a $\xi \to x \in P$.

Definícia 1. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. *Pravá vetná forma* v gramatike G je slovo $w \in (N \cup T)^*$ také, že $\sigma \Rightarrow^*_{rm} w$.

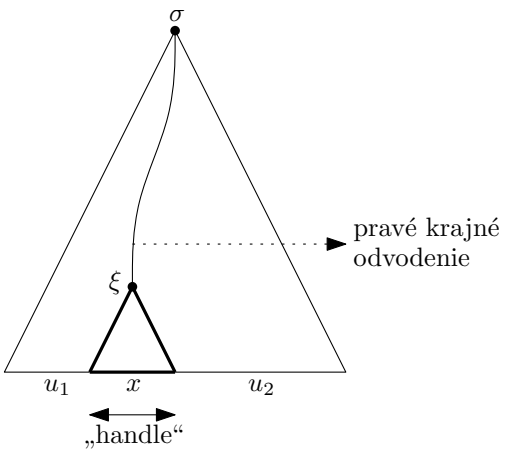
Definícia 2. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika a $w \in (N \cup T)^*$ je pravá vetná forma v gramatike G. Podslovo x pravej vetnej formy w sa nazýva „*handle*“[1,2], ak pre nejaké $u_1 \in (N \cup T)^*$, $u_2 \in T^*$ a $\xi \in N$ je

$$\sigma \Rightarrow^*_{rm} u_1 \xi u_2 \Rightarrow_{rm} u_1 x u_2,$$

pričom $w = u_1 x u_2$.

[1] Priamočiary slovenský preklad „rukoväť“ sa nezaužíval.

[2] Presnejšie za „handle“ považujeme *konkrétny výskyt* podslova x v pravej vetnej forme w spĺňajúci nasledujúcu podmienku.



Obr. 1: Podslovo x je „handle“ v pravej vetnej forme u_1xu_2. Keďže odvodenie na obrázku je pravé krajné, podslovo u_2 musí obsahovať výlučne terminálne symboly.

„Handle“ je teda podslovo x pravej vetnej formy w, ktorého zredukovaním podľa nejakého pravidla $\xi \to x$ vznikne predchádzajúca vetná forma v niektorom pravom krajnom odvodení vetnej formy w v gramatike G. Pojem „handle“ je ilustrovaný aj na obrázku 1, ktorý vysvetľuje voľbu tohto termínu – obrázok totiž nápadne pripomína rukoväť používanú na nosenie balíkov.

Algoritmy syntaktickej analýzy zdola nahor, ktoré sú založené na schéme „posuň a redukuj“ opísanej nižšie, zakaždým nájdu „handle“ v danom slove (presnejšie: nájdu „kandidátku na handle“, keďže ani nie je isté, či je dané slovo vetnou formou v G), ktorú zredukujú podľa niektorého pravidla, čím vznikne „kandidátka na predchádzajúcu vetnú formu“.

Ak je gramatika viacznačná – teda ak v nej existuje viac ako jedno pravé krajné odvodenie niektorého slova – môže jedna vetná forma w obsahovať aj viacero rôznych „handle“. V takom prípade ale ťažko očakávať, že by sa dala robiť syntaktická analýza zdola nahor deterministicky (a bez nutnosti úplného prehľadávania). Deterministické algoritmy teda budú narábať vždy len s určitými triedami *jednoznačných* gramatík. V jednoznačnej redukovanej gramatike obsahuje každá pravá vetná forma nanajvýš jednu „handle“ (presnejšie: vetná forma σ neobsahuje žiadnu „handle“ a ostatné pravé vetné formy obsahujú práve jednu „handle“).

2 Nedeterministická schéma algoritmu „posuň a redukuj“

Algoritmy syntaktickej analýzy zdola nahor častokrát pracujú na báze schémy „posuň a redukuj“ (angl. *„shift-reduce“*), ktorú teraz opíšeme. Táto schéma algoritmu je nedeterministická, nekladie žiadne obmedzenia na bezkontextovú gramatiku na vstupe a možno ju implementovať na zásobníkovom automate. Neskôr ukážeme, že pre isté triedy *špeciálnych* bezkontextových gramatík možno túto schému determinizovať a získať tak algoritmus v pravom slova zmysle.

Vstup: Bezkontextová gramatika $G = (N, T, P, \sigma)$ a slovo $w \in T^*$.

Schéma algoritmu:

- Nedeterministicky vykonaj jednu z nasledujúcich dvoch operácií:
  a) Zmaž z vrchu zásobníka nejaké slovo x a nahraď ho nejakým neterminálom ξ takým, že $\xi \to x \in P$. (*Redukuj* podľa $\xi \to x$.)
  b) Prečítaj jeden symbol zo vstupu a pridaj ho na zásobník. (*Posuň.*)
- Opakuj tento proces, až kým:
  a) Na vstupe nezostáva žiaden neprečítaný symbol a na zásobníku je počiatočná vetná forma σ. V takom prípade *akceptuj* ($w \in L(G)$).
  b) Nemožno vykonať ani operáciu „posuň“, ani operáciu „redukuj“.

Aspoň jedna vetva výpočtu takéhoto zásobníkového automatu na vstupe w akceptuje práve vtedy, keď $w \in L(G)$ – uvedenú schému tak možno chápať aj ako alternatívny dôkaz, že ku každej bezkontextovej gramatike existuje ekvivalentný zásobníkový automat. Ide o dôsledok skutočnosti, že jeho akceptačné vetvy „nájdu“ práve všetky *pravé krajné* odvodenia slova w. Ak napríklad

$$\sigma \Rightarrow_{rm} \alpha\beta \Rightarrow_{rm} \alpha b \Rightarrow_{rm} ab$$

je pravé krajné odvodenie, môže zásobníkový automat pracovať nasledovne („dno“ zásobníka označujeme symbolom $\vdash$, ktorého podobnosť s označením relácie kroku výpočtu v automate je čisto náhodná).

Zásobník	**Zvyšok vstupu**	**Operácia**
$\vdash$	ab	Posuň
$\vdash a$	b	Redukuj podľa $\alpha \to a$
$\vdash \alpha$	b	Posuň
$\vdash \alpha b$	ε	Redukuj podľa $\beta \to b$
$\vdash \alpha\beta$	ε	Redukuj podľa $\sigma \to \alpha\beta$
$\vdash \sigma$	ε	Akceptuj

Z tohto príkladu ľahko vidieť, že ku každému pravému krajnému odvodeniu

$$\sigma = w_0 \Rightarrow_{rm} w_1 \Rightarrow_{rm} w_2 \Rightarrow_{rm} \ldots \Rightarrow_{rm} w_m = w$$

v gramatike G zodpovedá nejaká akceptačná vetva výpočtu opísaného automatu – po zredukovaní „handle“ x pravej vetnej formy w_k pre $k \in [m]$ na neterminál ξ totiž na zásobníku bude nejaký prefix vetnej formy w_{k-1} končiaci neterminálom ξ a ľahko vidieť, že „handle“ vetnej formy w_{k-1} nemôže končiť pred týmto neterminálom. Na zásobník teda zo vstupu opäť možno posunúť niekoľko (prípadne aj nula) symbolov a získať tak na jeho vrchu „handle“, ktorú možno zredukovať. Každé odvodenie „nájdené“ opísaným automatom je naopak pravé krajné – ľahko totiž vidieť, že ak slovo x na vrchu zásobníka zredukujeme na neterminál ξ, nemôžu byť vo výslednej vetnej forme napravo od ξ žiadne ďalšie neterminály.

Poznámka 1. V každom kroku schémy „posuň a redukuj“ teda zreťazenie obsahu zásobníka a neprečítanej časti vstupu zodpovedá aktuálnej „kandidátke na vetnú formu“ gramatiky G, pričom na zásobníku je po každej redukcii jej prefix, ktorý ešte v hľadanom (a od konca konštruovanom) pravom krajnom odvodení nebol „sterminálnený“. Ak je navyše gramatika jednoznačná a redukovaná, môže byť v akceptačnej vetve na zásobníku prefix siahajúci najviac po koniec (jedinej) „handle“. Pre viacznačné gramatiky je zas v každej akceptačnej vetve vždy na zásobníku prefix siahajúci najviac po koniec *niektorej* „handle“. Takéto prefixy pravých vetných foriem sa zvyknú nazývať *životaschopné prefixy* (angl. *viable prefixes*).[3] S týmto pojmom budeme pracovať v druhej časti týchto poznámok, v súvislosti s LR gramatikami.

V praxi je nedeterministická schéma „posuň a redukuj“ užitočná len málo, keďže jej determinizácia preskúmaním všetkých nedeterministických vetiev je neefektívna. Preto sa skúmali predovšetkým *triedy gramatík, pre ktoré možno schému „posuň a redukuj“ realizovať deterministicky* – v každom kroku musí byť zrejmé, či posunúť alebo redukovať a ak redukovať, musí byť zrejmé, podľa ktorého pravidla. Musí teda existovať kritérium, pomocou ktorého možno deterministicky rozoznať situáciu, keď je možné, že na vrchu zásobníka je „handle“. V takom prípade musí byť táto „handle“ x daná jednoznačne a nesmie byť možné, že na zásobníku vznikne „handle“ až po posunutí symbolu zo vstupu. Navyše musí existovať práve jeden neterminál ξ, na ktorý možno túto „handle“ redukovať.

V nasledujúcom sa budeme zaoberať *konkrétnymi triedami* takýchto gramatík. Najprv preskúmame tzv. *jednoducho precedenčné gramatiky*, pre ktoré je možné rozhodnúť medzi operáciou „posuň“ a operáciou „redukuj“ podľa dvojíc po sebe idúcich symbolov v slove. Na ďalšom cvičení sa potom budeme zaoberať o niečo zložitejšou a omnoho dôležitejšou triedou gramatík, tzv. LR gramatikami.

[3] Definície životaschopných prefixov sa pre viacznačné gramatiky môžu líšiť; podstatná je ale iba ich definícia pre jednoznačné gramatiky.

3 Jednoducho precedenčné gramatiky

Jednoducho precedenčné gramatiky sú bezkontextové gramatiky, ktorých symboly spĺňajú isté podmienky umožňujúce v schéme „posuň a redukuj“ na základe dvojice po sebe idúcich písmen rozhodnúť, či posunúť alebo redukovať. V prípade redukovania je navyše možné deterministicky určiť „kandidátku na handle“, ktorú treba zredukovať, ako aj neterminál, ktorým je potrebné toto redukované podslovo na zásobníku nahradiť.

Poznámka 2. Determinizácia schémy „posuň a redukuj“ bude pre jednoducho precedenčné gramatiky možná iba za predpokladu, že jej vstup pozostáva zo slova w nasledovaného špeciálnym symbolom $\dashv$ označujúcim koniec vstupného slova. Tento drobný technický detail je ale zaujímavý iba z teoretického hľadiska – v praxi nie je s pridaním symbolu $\dashv$ očividne žiaden problém.

V druhej časti týchto poznámok budeme pracovať s tzv. LR(0) gramatikami, pre ktoré tento predpoklad nie je nutný.

Pri definícii jednoducho precedenčných gramatík budeme využívať tri špeciálne binárne relácie na abecede $N \cup T$, ktoré budeme označovať symbolmi $\doteq$, $\lessdot$ a $\gtrdot$. Vysvetlime si najprv približný intuitívny význam týchto relácií pre gramatiku $G = (N, T, P, \sigma)$:

$\doteq$: Pre $a, b \in N \cup T$ je $a \doteq b$ práve vtedy, keď „je možné“, že v gramatike G existuje pravá vetná forma $w = xyz$ s „handle“ y takou, že $y = y_1 a b y_2$ pre nejaké slová y_1, y_2. Stručnejšie povedané: $a \doteq b$, ak „je možné“, že sa symboly a a b môžu vyskytovať za sebou v „handle“.

$\lessdot$: Pre $a, b \in N \cup T$ je $a \lessdot b$ práve vtedy, keď „je možné“, že v gramatike G existuje pravá vetná forma $w = xyz$ s „handle“ y taká, že $x = x'a$ a $y = by'$ pre nejaké slová x', y'. Stručnejšie povedané: $a \lessdot b$, ak „je možné“, že sa symbol a môže vyskytovať ako posledný symbol pred „handle“ začínajúcou symbolom b.

$\gtrdot$: Pre $a, b \in N \cup T$ je $a \gtrdot b$ práve vtedy, keď „je možné“, že v gramatike G existuje pravá vetná forma $w = xyz$ s „handle“ y taká, že $y = y'a$ a $z = bz'$ pre nejaké slová y', z'. Stručnejšie povedané: $a \gtrdot b$, ak „je možné“, že sa symbol a môže vyskytovať ako posledný symbol v „handle“, ktorá je vo vetnej forme nasledovaná symbolom b.

Poznámka 3. Mierna nepresnosť tejto charakterizácie relácií $\doteq$, $\lessdot$ a $\gtrdot$ spočíva vo vágnosti slovného spojenia „je možné“, bez ktorého by išlo len o implikácie sprava doľava a nie o ekvivalencie.

Ak pre dvojicu symbolov a, b platí niektoré z tvrdení, tak sú tieto symboly v zodpovedajúcej relácii. Relácie $\doteq$, $\lessdot$ a $\gtrdot$ však formálne definujeme so zreteľom na ich efektívnu „zostrojiteľnosť“, a preto dvojica symbolov bude môcť byť v niektorej z relácií aj bez toho, aby pre ňu platilo príslušné tvrdenie. Ak napríklad gramatika G nie je redukovaná a obsahuje pravidlo $\xi \to ab$ také, že symboly a a b sa nemôžu vyskytovať v žiadnej vetnej forme gramatiky G, stále bude platiť $a \doteq b$, hoci sa tieto dva symboly očividne nemôžu vyskytovať za sebou v žiadnej „handle“ (pretože „handle“ je definovaná iba pre pravé vetné formy).

Pristúpme teraz k naozajstnej formálnej definícii relácií $\doteq$, $\lessdot$ a $\gtrdot$. Za týmto účelom najprv zavedieme pomocné relácie FIRST a LAST.

Definícia 3. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. Potom:

a) FIRST je binárna relácia na $N \cup T$ taká, že pre všetky $a, b \in N \cup T$ je $(a, b) \in \text{FIRST}$ práve vtedy, keď $a \in N$ a súčasne existuje $x \in (N \cup T)^*$ také, že $a \to bx \in P$.

b) LAST je binárna relácia na $N \cup T$ taká, že pre všetky $a, b \in N \cup T$ je $(a, b) \in \text{LAST}$ práve vtedy, keď $a \in N$ a súčasne existuje $x \in (N \cup T)^*$ také, že $a \to xb \in P$.

Poznámka 4. Ako FIRST^+, LAST^+, FIRST^* a LAST^* označujeme tranzitívne resp. reflexívne a tranzitívne uzávery relácií FIRST a LAST. Pre „bezepsilonovú“ gramatiku je teda dvojica (a, b) v relácii FIRST^+ práve vtedy, keď zo symbolu a možno na nenulový počet krokov odvodiť slovo začínajúce na b. V relácii FIRST^* sú navyše aj všetky dvojice (a, a) pre $a \in N \cup T$.

Definícia 4. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. Potom:

a) $\doteq$ je binárna relácia na $N \cup T$ taká, že pre všetky $a, b \in N \cup T$ je $a \doteq b$ práve vtedy, keď existuje neterminál $\xi \in N$ a slová $u, v \in (N \cup T)^*$ také, že $\xi \to uabv \in P$.

b) $\lessdot$ je binárna relácia na $N \cup T$ taká, že pre všetky $a, b \in N \cup T$ je $a \lessdot b$ práve vtedy, keď existujú $\xi, \eta \in N$ a $u, v \in (N \cup T)^*$ také, že $\xi \to ua\eta v \in P$ a $(\eta, b) \in \mathrm{FIRST}^+$.

c) $\gtrdot$ je binárna relácia na $N \cup T$ taká, že pre všetky $a, b \in N \cup T$ je $a \gtrdot b$ práve vtedy, keď $b \in T$ a existujú $\xi, \eta \in N$, $c \in N \cup T$ a $u, v \in (N \cup T)^*$ také, že $\xi \to u\eta cv \in P$, $(\eta, a) \in \mathrm{LAST}^+$ a $(c, b) \in \mathrm{FIRST}^*$.

Neskôr v týchto poznámkach ukážeme, že takto definované relácie $\doteq$, $\lessdot$ a $\gtrdot$ možno skutočne pre každú bezkontextovú gramatiku efektívne vypočítať.

Poznámka 5. Overíme teraz, že uvedené definície skutočne korešpondujú s intuitívnym vysvetlením významu relácií $\doteq$, $\lessdot$ a $\gtrdot$ z úvodu tohto oddielu. Ak teda v týchto intuitívnych vysvetleniach upustíme od vágneho slovného spojenia „je možné“ a ekvivalenciu nahradíme implikáciou sprava doľava, dostaneme pravdivé tvrdenia.

$\doteq$: Nech sa a a b môžu vyskytovať za sebou v „handle“. Z definície „handle“ vyplýva, že symboly a a b sa musia vyskytovať za sebou aj na pravej strane niektorého pravidla, a teda $a \doteq b$.

$\lessdot$: Nech sa a môže vyskytovať ako posledný symbol pred „handle“ začínajúcou symbolom b. Takáto pravá vetná forma je teda tvaru $x'aby'z$, kde $y = by'$ je „handle“. Pravé krajné odvodenie vetnej formy $x'aby'z$ je potom tvaru

$$\sigma \Rightarrow_{rm}^* x'a\beta z \Rightarrow_{rm} x'aby'z.$$

Symbol a vo vetnej forme $x'a\beta z$ musel vzniknúť použitím nejakého pravidla $\alpha \to ua\eta v \in P$ takého, že $(\eta, \beta) \in \mathrm{FIRST}^*$.[4] Keďže navyše $(\beta, b) \in \mathrm{FIRST}$ (lebo z uvedeného odvodenia vyplýva $\beta \to by' \in P$), je $(\eta, b) \in \mathrm{FIRST}^+$, a teda aj $a \lessdot b$.

$\gtrdot$: Nech sa a môže vyskytovať ako posledný symbol v „handle“, ktorá je vo vetnej forme nasledovaná symbolom b. Takáto pravá vetná forma je tvaru $xy'abz'$, kde $y = y'a$ je „handle“. Pravé krajné odvodenie vetnej formy $xy'abz'$ je potom tvaru

$$\sigma \Rightarrow_{rm}^* x\alpha bz' \Rightarrow_{rm} xy'abz'.$$

Symbol b je nutne terminál. Ak by totiž bol b neterminál, musel by sa v pravom krajnom odvodení prepísať skôr, než neterminál α. Nech ψ je prvý spoločný predok symbolov α a b v danom pravom krajnom odvodení. Potom musí existovať pravidlo $\psi \to u\eta cv \in P$, kde $(\eta, \alpha) \in \mathrm{LAST}^*$ a $(c, b) \in \mathrm{FIRST}^*$. Keďže navyše $(\alpha, a) \in \mathrm{LAST}$ (lebo nutne $\alpha \to y'a \in P$), je $(\eta, a) \in \mathrm{LAST}^+$, a teda $a \gtrdot b$.

V prípade, že sú každé dva symboly z $N \cup T$ v najviac jednej z relácií $\doteq$, $\lessdot$ a $\gtrdot$, možno tieto relácie za určitých ďalších predpokladov využiť na determinizáciu algoritmu „posuň a redukuj“. To je odzrkadlené v nasledujúcej definícii jednoducho precedenčnej gramatiky.

Definícia 5. Bezkontextová gramatika $G = (N, T, P, \sigma)$ je *jednoducho precedenčná*, ak sú splnené nasledujúce podmienky:

(i) Gramatika G je „bezepsilonová“, t. j. $P \subseteq N \times (N \cup T)^+$.

(ii) Gramatika G je spätne deterministická, t. j. pre každé $x \in (N \cup T)^*$ existuje najviac jeden neterminál $\xi \in N$ taký, že $\xi \to x \in P$.

(iii) Každá dvojica symbolov $(a, b) \in (N \cup T)^2$ je v najviac jednej z relácií $\doteq$, $\lessdot$ a $\gtrdot$.

[4] V opačnom prípade by musel vzniknúť na nenulový počet krokov z neterminálu ψ takého, že $\alpha \to u\psi\eta v \in P$. V pravom krajnom odvodení ale nemožno prepísať neterminál ψ skôr, než sa „sterminálni“ η, čo znamená, že v takomto prípade nemožno odvodiť vetnú formu $x'a\beta z$.

Poznámka 6. Môže sa stať, že v jednoducho precedenčnej gramatike $G = (N, T, P, \sigma)$ nejaké dva symboly $a, b \in N \cup T$ nie sú *v ani jednej* z relácií $\doteq$, $\lessdot$ a $\gtrdot$. To znamená, že symboly a, b sa nemôžu vyskytovať vedľa seba v žiadnej pravej vetnej forme gramatiky G. Inak by totiž nejaká vetná forma obsahujúca podslovo ab musela byť „zredukovateľná“ na počiatočný neterminál σ a v takom prípade možno ľahko nahliadnuť, že symboly a, b musia byť v aspoň jednej z relácií $\doteq$, $\lessdot$ a $\gtrdot$.

Pre jednoducho precedenčné gramatiky je možné v schéme „posuň a redukuj“ v každom kroku deterministicky rozhodnúť, či vykonať operáciu „posuň“ alebo operáciu „redukuj“. Nech a je symbol na vrchu zásobníka – čiže posledný symbol prefixu spracúvaného na zásobníku – a b je prvý symbol zvyšku slova. Ak $a \doteq b$ alebo $a \lessdot b$, na danej pozícii v slove nemôže končiť „handle“ a jedinou možnosťou je teda posunúť symbol zo vstupu. Ak $a \gtrdot b$, jedinou možnosťou je redukovať podľa nejakého pravidla, ktorého pravá strana sa končí symbolom a, keďže na danej pozícii nemôže žiadna „handle“ pokračovať ani začínať. Začiatok redukovaného podslova v takom prípade možno nájsť podľa relácií medzi symbolmi postupne odoberanými zo zásobníka. Neterminál, na ktorý sa toto podslovo zredukuje, je jednoznačne určený vďaka spätnému determinizmu gramatiky G. V prípade, že na zásobníku nie je žiaden symbol (okrem špeciálneho symbolu $\vdash$ pre jeho „dno“), je jedinou možnosťou posunúť symbol zo vstupu. Podobne v prípade, že na vstupe nie je žiaden neprečítaný symbol (okrem špeciálneho symbolu $\dashv$ pre koniec vstupného slova, ktorý je podľa predpokladu z poznámky 2 súčasťou vstupu), je jedinou možnosťou redukovať (pokiaľ existuje vhodné pravidlo).

Algoritmus „posuň a redukuj“ pre jednoducho precedenčné gramatiky teda možno opísať nasledovne (zrejmé detaily súvisiace s inicializáciou vynechávame):

Vstup: Jednoducho precedenčná gramatika $G = (N, T, P, \sigma)$ a slovo $w \dashv$, kde $w \in T^*$.

Algoritmus:

1. Ak je na vrchu zásobníka $\vdash$, *posuň* symbol zo vstupu a pokračuj krokom 5.
2. Ak je na vrchu zásobníka symbol a a prvý neprečítaný symbol na vstupe je b, pričom $a \doteq b$, *posuň* symbol zo vstupu a pokračuj krokom 5.
3. Ak je na vrchu zásobníka symbol a a prvý neprečítaný symbol na vstupe je b, pričom $a \lessdot b$, *posuň* symbol zo vstupu a pokračuj krokom 5.
4. Ak je na vrchu zásobníka symbol a a prvý neprečítaný symbol na vstupe je b, pričom $a \gtrdot b$, *redukuj*. *Redukuj* aj v prípade, že na vstupe je $\dashv$ a na zásobníku je aspoň jeden symbol (okrem $\vdash$). Začiatok redukovaného podslova pritom hľadaj nasledovne:
   a) Odober symbol c z vrchu zásobníka.
   b) Nech d je nový vrch zásobníka. V takomto prípade nemôže byť $d \gtrdot c$ (prečo?).
   c) Ak $d \doteq c$, opakuj celý proces počnúc krokom a).
   d) Ak $d \lessdot c$ alebo $d = \vdash$, slovo doposiaľ zmazané zo zásobníka je hľadané redukované podslovo y.

   Vďaka spätnému determinizmu existuje *najviac jeden* neterminál ξ taký, že $\xi \to y \in P$. Ak takýto neterminál ξ existuje, pridaj ξ na zásobník.
5. Opakuj celý tento proces počnúc krokom 1, až kým:
   a) Na vstupe nie je žiaden neprečítaný symbol (okrem $\dashv$) a na zásobníku je nad jeho „dnom“ $\vdash$ počiatočná vetná forma σ. V takom prípade *akceptuj* – $w \in L(G)$.
   b) Nemožno vykonať ani operáciu „posuň“ ani operáciu „redukuj“. V takom prípade *zamietni* – $w \notin L(G)$.

Príklad použitia uvedeného algoritmu možno nájsť v závere týchto poznámok.

4 Výpočet relácií $\doteq$, $\lessdot$ a $\gtrdot$ pre danú gramatiku

V nasledujúcom ukážeme, ako pre danú bezkontextovú gramatiku G algoritmicky vypočítať relácie $\doteq$, $\lessdot$ a $\gtrdot$. Takto vypočítané relácie možno využiť na overenie, či je daná gramatika G jednoducho precedenčná a ak je, aj pri vykonávaní algoritmu „posuň a redukuj“. Pre jednu gramatiku G je očividne postačujúce vypočítať relácie $\doteq$, $\lessdot$ a $\gtrdot$ iba raz (nezávisia totiž od vstupu).

Relácie FIRST a LAST možno očividne vypočítať priamo z ich definície. Ich tranzitívny resp. reflexívny a tranzitívny uzáver potom možno vypočítať štandardným spôsobom. Napríklad pre výpočet FIRST^+ to znamená postupne „generovať“ relácie

$$\mathrm{FIRST}, \bigcup_{i=1}^{2} \mathrm{FIRST}^i, \bigcup_{i=1}^{3} \mathrm{FIRST}^i, \ldots,$$

až kým sa nebudú niektoré dve z týchto relácií rovnať (alebo použiť nejakú menej „drevorubačskú“ metódu; napríklad možno využiť násobenie matíc). Pre reflexívny a tranzitívny uzáver treba ešte reláciu zjednotiť s identickou reláciou.

Reláciu $\doteq$ možno taktiež vypočítať priamo z definície – sú v nej všetky dvojice symbolov, ktoré sa vyskytujú za sebou na pravej strane niektorého pravidla.

Nasledujúce tvrdenia, kde RS označuje štandardnú kompozíciu relácií R a S, sú priamym dôsledkom definícií relácií $\lessdot$ a $\gtrdot$.

Tvrdenie 1. *Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. Potom sa relácia $\lessdot$ rovná relácii*

$$\doteq \mathrm{FIRST}^+.$$

Tvrdenie 2. *Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika. Potom sa relácia $\gtrdot$ rovná relácii*

$$\left((\mathrm{LAST}^+)^{-1} \doteq \mathrm{FIRST}^* \right) \cap ((N \cup T) \times T).$$

Poznámka 7. Vypočítať inverznú reláciu k relácii LAST^+ je triviálnou záležitosťou – stačí iba „otočiť“ všetky dvojice.

5 Príklad

V nasledujúcom demonštrujeme metódy z predchádzajúcich oddielov na príklade bezkontextovej gramatiky G, pre ktorú:

a) Vypočítame relácie $\doteq$, $\lessdot$ a $\gtrdot$.

b) Overíme, že G je jednoducho precedenčná.

c) Odsimulujeme algoritmus „posuň a redukuj“ na konkrétnom vstupe w.

Príklad 1. Nech $G = (N, T, P, \sigma)$ je bezkontextová gramatika s $N = \{\sigma, \alpha, \beta\}$, $T = \{a, b, c, d\}$ a

$$\begin{aligned} P = \{ \sigma &\to a\alpha \mid \beta d \\ \alpha &\to c \mid ca \\ \beta &\to \beta b \mid ba \}. \end{aligned}$$

Vypočítajme najprv relácie FIRST a LAST. Zrejme

$$\begin{aligned} \mathrm{FIRST} &= \{(\sigma, a), (\sigma, \beta), (\alpha, c), (\beta, \beta), (\beta, b)\}, \\ \mathrm{LAST} &= \{(\sigma, d), (\sigma, \alpha), (\alpha, a), (\alpha, c), (\beta, a), (\beta, b)\}. \end{aligned}$$

Tranzitívne uzávery relácií FIRST a LAST vypočítame nasledovne:

$$\begin{aligned}
\mathrm{FIRST}^1 &= \{(\sigma,a),(\sigma,\beta),(\alpha,c),(\beta,\beta),(\beta,b)\},\\
\mathrm{FIRST}^1 \cup \mathrm{FIRST}^2 &= \{(\sigma,a),(\sigma,b),(\sigma,\beta),(\alpha,c),(\beta,\beta),(\beta,b)\},\\
\mathrm{FIRST}^1 \cup \mathrm{FIRST}^2 \cup \mathrm{FIRST}^3 &= \mathrm{FIRST}^1 \cup \mathrm{FIRST}^2 = \mathrm{FIRST}^+.
\end{aligned}$$

a

$$\begin{aligned}
\mathrm{LAST}^1 &= \{(\sigma,d),(\sigma,\alpha),(\alpha,a),(\alpha,c),(\beta,a),(\beta,b)\},\\
\mathrm{LAST}^1 \cup \mathrm{LAST}^2 &= \{(\sigma,a),(\sigma,c),(\sigma,d),(\sigma,\alpha),(\alpha,a),(\alpha,c),(\beta,a),(\beta,b)\},\\
\mathrm{LAST}^1 \cup \mathrm{LAST}^2 \cup \mathrm{LAST}^3 &= \mathrm{LAST}^1 \cup \mathrm{LAST}^2 = \mathrm{LAST}^+.
\end{aligned}$$

Reflexívny a tranzitívny uzáver FIRST^* je daný ako $\mathrm{FIRST}^* = I \cup \mathrm{FIRST}^+$, kde I označuje identickú reláciu. Preto

$$\mathrm{FIRST}^* = \{(\sigma,\sigma),(\alpha,\alpha),(\beta,\beta),(a,a),(b,b),(c,c),(d,d),(\sigma,a),(\sigma,b),(\sigma,\beta),(\alpha,c),(\beta,b)\}.$$

Zjavne tiež

$$(\mathrm{LAST}^+)^{-1} = \{(a,\sigma),(c,\sigma),(d,\sigma),(\alpha,\sigma),(a,\alpha),(c,\alpha),(a,\beta),(b,\beta)\}.$$

Takto vypočítané pomocné relácie teraz môžeme využiť na výpočet samotných relácií $\doteq$, $\lessdot$ a $\gtrdot$. Reláciu $\doteq$ ešte možno vypočítať priamo z jej definície:

$$\doteq \text{ obsahuje dvojice } \{(a,\alpha),(\beta,d),(c,a),(\beta,b),(b,a)\}.$$

Relácia

$$\lessdot \text{ je daná ako } \doteq \mathrm{FIRST}^+, \text{ t. j. } \{(a,c)\}.$$

Napokon relácia

$$\gtrdot \text{ je daná ako } \left((\mathrm{LAST}^+)^{-1} \doteq \mathrm{FIRST}^*\right) \cap ((N \cup T) \times T)\,, \text{ t. j. } \{(a,b),(a,d),(b,b),(b,d)\}.$$

V nasledujúcom overíme, že gramatika G je jednoducho precedenčná. Postupne teda ukážeme, že G spĺňa podmienky (i) až (iii) z definície 5.

(i) Gramatika G je „bezepsilonová“, pretože v P nie je žiadne pravidlo typu $\xi \to \varepsilon$.

(ii) Gramatika G je spätne deterministická, pretože pre všetky $x \in (N \cup T)^*$ existuje najviac jeden neterminál $\xi \in N$ taký, že $\xi \to x \in P$.

(iii) Relácie $\doteq$, $\lessdot$ a $\gtrdot$, ktoré sme vypočítali vyššie, možno zapísať v podobe tabuľky nasledovne.

	σ	α	β	a	b	c	d
σ							
α							
β					$\doteq$		$\doteq$
a		$\doteq$			$\gtrdot$	$\lessdot$	$\gtrdot$
b				$\doteq$	$\gtrdot$		$\gtrdot$
c				$\doteq$			
d							

Ľahko vidieť, že každá dvojica symbolov z $N \cup T$ je v najviac jednej z relácií $\doteq$, $\lessdot$ a $\gtrdot$.

Gramatika G teda skutočne *je* jednoducho precedenčná. Odsimulujeme teraz algoritmus „posuň a redukuj“ pre gramatiku G na vstupe $w = babbd$.

Zásobník	**Zvyšok vstupu**	**Operácia**
$\vdash$	$babbd \dashv$	Posuň ($\vdash$ na zásobníku)
$\vdash b$	$abbd \dashv$	Posuň ($b \doteq a$)
$\vdash ba$	$bbd \dashv$	Redukuj ($a \gtrdot b$) podľa $\beta \to ba$
$\vdash \beta$	$bbd \dashv$	Posuň ($\beta \doteq b$)
$\vdash \beta b$	$bd \dashv$	Redukuj ($b \gtrdot b$) podľa $\beta \to \beta b$
$\vdash \beta$	$bd \dashv$	Posuň ($\beta \doteq b$)
$\vdash \beta b$	$d \dashv$	Redukuj ($b \gtrdot d$) podľa $\beta \to \beta b$
$\vdash \beta$	$d \dashv$	Posuň ($\beta \doteq d$)
$\vdash \beta d$	$\dashv$	Redukuj ($\dashv$ na vstupe) podľa $\sigma \to \beta d$
$\vdash \sigma$	$\dashv$	Akceptuj (σ na zásobníku, dočítaný vstup)

Pri redukcii nájde algoritmus redukovanú vetnú formu podľa relácií medzi symbolmi na zásobníku. Napríklad pri prvej redukcii je na zásobníku ba, pričom $b \doteq a$. Algoritmus „posuň a redukuj“ teda vyprázdni celý zásobník bez toho, aby „narazil“ na dvojicu symbolov v relácii $\lessdot$. Preto treba redukovať celý obsah zásobníka ba. Zodpovedajúca ľavá strana pravidla β je jednoznačne určená vďaka spätnému determinizmu.

Príklad 2. Uvažujme teraz jemne pozmenenú gramatiku $G' = (N', T', P', \sigma')$, kde $N' = \{\sigma, \alpha, \beta\}$, $T' = \{a, b, c, d\}$, $\sigma' = \sigma$ a

$$\begin{aligned} P' = \{&\sigma \to a\alpha \mid \beta b \\ &\alpha \to c \mid ca \\ &\beta \to \beta b \mid ba\}. \end{aligned}$$

O tejto gramatike možno okamžite konštatovať, že nie je jednoducho precedenčná, pretože obsahuje súčasne pravidlá $\sigma \to \beta b$ a $\beta \to \beta b$. Gramatika G' teda nie je spätne deterministická.

Jednoducho precedenčná nie je ani bezkontextová gramatika $G'' = (N'', T'', P'', \sigma'')$ taká, že $N'' = \{\sigma, \alpha, \beta\}$, $T'' = \{a, b, c, d\}$, $\sigma'' = \sigma$ a

$$\begin{aligned} P'' = \{&\sigma \to a\alpha \mid \varepsilon \\ &\alpha \to c \mid ca \\ &\beta \to \beta b \mid ba\}. \end{aligned}$$

Obsahuje totiž pravidlo $\sigma \to \varepsilon$, a teda nie je „bezepsilonová“.

Poznámka 8. Symboly $\doteq$, $\lessdot$ a $\gtrdot$ môžu evokovať určité vlastnosti niektorých relácií z matematiky, ako napríklad $=$, $<$ a $>$. Treba preto dôrazne upozorniť na fakt, že relácie $\doteq$, $\lessdot$ a $\gtrdot$ nemajú „takmer žiadnu z pekných vlastností“ týchto známych relácií. Relácia $\doteq$ napríklad nemusí byť symetrická, ako ukazuje príklad vyššie. Podobne môže platiť $a \lessdot b$ bez toho, aby platilo $b \gtrdot a$. Dvojica symbolov (a, b) tiež môže byť vo všetkých troch reláciách $\doteq$, $\lessdot$ a $\gtrdot$ súčasne.

Poznámka 9. Symboly $\vdash$ a $\dashv$ nepatria ani do množiny neterminálov, ani do množiny terminálov jednoduchej precedenčnej gramatiky – ide iba o pomocné symboly zavedené v súvislosti s algoritmom „posuň a redukuj“. Preto pre tieto symboly nie je definovaná žiadna z relácií $\doteq$, $\lessdot$ a $\gtrdot$. Napriek tomu možno na tieto symboly intuitívne nazerať tak, ako keby pre každé $c \in N \cup T$ bolo $\vdash \lessdot c$ a $c \gtrdot \dashv$. Operácie vykonávané algoritmom „posuň a redukuj“ v špeciálnych prípadoch (keď na zásobníku je $\vdash$ alebo keď na vstupe je $\dashv$) sú potom konzistentné so zvyšnými prípadmi.