

Формални Езици и Езикови Процесори
ТУ, кат. КС, летен семестър 2012

Лекция 15

Тема:

Синтактичен Анализ (Операторно Предшествие)

Съдържание:

- Синтактичен Анализ
- Формални Методи
- Възходяща /Bottom-Up/ Стратегия
- Метод на операторно предшествие /operator precedence method/
- Алгоритъм за конвертиране на АИ от инфиксен запис в обратен полски запис
- Демо програма

СА – формални методи

Дадено: КСГ граматика G , описваща език $L(G)$ и входен низ ω .

Задача: Изречение ли е вх. низ в езика $L(G)$?

Принадлежи ли низът на езика $\omega \in L(G)$?

Ако Да, да се построи syntax (parse) tree.

Иначе, грешка

Две стратегии за строеж на синтактично дърво:

Низходяща top-down (from root to leaves);

Възходяща bottom-up (from leaves to root).

Bottom-up възходящ подход

Постановка :

Входна лента

Изх. лента

Стек

Статус преди СА

Статус след СА

скан показалец

скан показалец

top stack: празен стек

старт. нетерминал

Припомням: основа – най-лява проста фраза

СА – формален проблем

Дадено: КСГ граматика G , описваща език $L(G)$ и входен низ ω .

Задача: Изречение ли е вх. низ в езика $L(G)$?
Принадлежи ли низът на езика $\omega \in L(G)$?

Ако Да, да се построи syntax (parse) tree.

Иначе, грешка

Как се строи синт. дърво? В каква форма се представя?
Генериране и съхранение на продукции такива, че приложени в прав ред ще потвърдят пораждането/деривацията от S до ω или приложени в обратен ред ще редуцират/свият входния низ от ω до стартовия нетерминал S .

СА като функция

Дадено: КСГ граматика G , описваща език $L(G)$ и входен низ string ω .

Задачата за СА представена като функция:

$$SA(\omega) = k_1 k_2 k_3 \dots k_n$$

Вх. Низ ω е аргумент на функцията.

Функцията SA връща последователност от номера на продукции (т.е абстрактна форма на дървото на разбора).

Пример: Bottom-up СА на АИ

$$a + a * a$$

Дадено: еднозначна КСГ граматика на АИ

1. $E \rightarrow T$

2. $E \rightarrow E + T$

3. $T \rightarrow F$

4. $T \rightarrow T * F$

5. $F \rightarrow a$

6. $F \rightarrow (E)$

$$\underline{a} + a * a > \underline{F} + a * a > \underline{T} + a * a > E + \underline{a} * a > \\ E + \underline{F} * a > E + T * \underline{a} > E + \underline{T} * F > \underline{E + T} > E$$

$$\text{СА } (a+a*a) = 5 3 1 5 3 5 4 2$$

5/20/2012 Десен каноничен анализ в обратен ред

основа – най-лява проста фраза

Bottom-up СА методи са известни като SR-методи.

- SR – Shift (входният символ се записва shift/push в стека)
- SR – Reduce (на върха на стека се локализира дясна страна на правило, която се редуцира/свива до съответния ляв нетерминал)
- Подниз на СФ (дясна страна на продукцията) локализиран и подлежащ на редукция до ляв нетерминал се нарича *основа/связка/handle* – най-лявата проста фраза в една СФ

Основа – формална дефиниция

Основата като най-лява проста фраза не винаги е физически най-лявата проста фраза.

... $\underline{E+T} * F > E * F$ – грешка, няма път към E

... $E + \underline{T * F} > \underline{E+T} > E$ – успех

Handle на СФ е най-левият низ проста фраза такава, че след свиването ѝ до левия нетерминал, отново валидна СФ се получава.

Затова, за да се открие основа не е достатъчно да се локализира единствено проста фраза.

Нужен/необходим е анализ на контекста.

Нов пример

Дадени граматика и низ за разпознаване: $aaba$

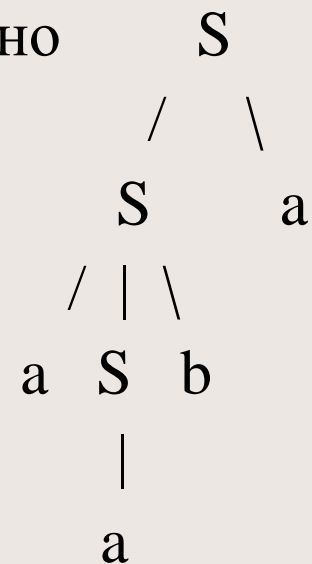
Нека дървото на разбора е построено

$S \rightarrow S a$

$S \rightarrow a S b$

$S \rightarrow a$

$S \rightarrow b$



Последователността от редукции е показана:

$a \underline{a} b a > \underline{a} S b a > \underline{S} a > S$

Основа/Handle (прод.)

- Всички възходящи СА методи са базирани на цикъл, в тялото на който се локализира основа с последваща редукция до ляв нетерминал.
- При излизане от цикъла, стартовият нетерминал се очаква да е единственият символ на върха на стека.
- Ако се излезе от цикъла в друг контекст или не може да се локализира основа, това е грешка.
- Теорема: **Надясно от основата има само терминални символи.**

Възходящи СА методи

- a/ Определяне на основата чрез ограничен контекст (Bound Context $BC(m,n)$ граматика);
- b/ Определяне на основата чрез отношения на просто предшествие между терминалните и нетерминалните символи (граматика с просто предшествие);
- c/ **Определяне на основата чрез отношения на операторно предшествие между терминалните символи (граматика с операторно предшествие);**
- d/ Определяне на основата чрез LR ситуации (LR(k) граматика, LR(1) граматика, SLR(1) граматика, LALR(1) граматика).

Метод ПП просто предшествие

Wirth и Weber (1966) откриха СА метод, който определя основата чрез сравнение на двойка символи (top stack char - current input char, или два съседни stack chars). Отношения, наречени просто предшествие $<$, $=$, $>$ се въвеждат за сравнение на двойката символи.

Отношения на просто предшествие се дефинират между терминални и нетерминални символи

Следният алгоритъм служи да локализира основата::

Даден е подниз от СФ: ... $s_1 s_2 s_3 \dots s_4 s_5 s_6 \dots$

Ако се установи конфигурация с ПП отношения такава, че $s_1 < s_2$, $s_2 = s_3$, ... , $s_4 = s_5$, $s_5 > s_6$ тогава поднизът $s_2 s_3 \dots s_4 s_5$ се класифицира като основа/handle.

Метод ПП просто предшествие

С други думи:

- ПП релация $<$ е ляв ограничител на основата.
- ПП релация $>$ е десен ограничител на основата.
- ПП релация $=$ е валидна между съседните символи от основата.
- Следват примери с ПП релации.

Релации ПП просто предшествование

Определение:

$$X = Y, \text{ iff}$$

Еднакво предшествование за X и Y

$$X < Y, \text{ iff}$$

Y предшества X (X е ляв ограничител)

$$X > Y, \text{ iff}$$

X предшества Y (Y е ограничител отдясно)

Критика към СА ПП метод

- Т.к ПП релациите са възможни между всички терминали и/или нетерминали, размерът на упр таблица се получава твърде голям, i.e.
$$\text{row}(\Sigma \cup N) \times \text{row}(\Sigma \cup N)$$
- С цел редуциране размера на упр таблица, се въвежда модификация на СА ПП метод, наречен СА метод на *операторно предшествие*, базиран на същия тип отношения ОП $<, =, >$ които са възможни само между терминални символи

Метод ОП операторно предшествие

ОП СА метод се прилага върху операторни граматика.

Определение: Една КСГ граматика е операторна, ако:

a/ няма ϵ -продукции;

b/ няма продукции от вида $V \rightarrow \alpha X Y \beta$,
 $X, Y \in N$, т.е без съседни нетерминали в дясна страна на продукциите.

Релации ОП операторно предшествие

Определение:

$a = b$, iff

еднакво операторно предшествие за a и b

$a < b$, iff

b предшества a (a е ляв ограничител)

$a > b$, iff

a предшества b (b е ограничител отдясно)

Как се определя основата

локализиране (фиксиране) на
основата чрез отношения на
предшествие (просто предшествие
и операторно предшествие)

< ляв ограничител на основата

= еднакво предшествие

> десен ограничител на основата

Метод ОП операторно предшествие

Една ОП граматика се определя като операторна КСГ граматика с:

а/ не повече от едно отношение на ОП (т.е 0 или 1) в коя да е двойка символи, съставена от терминални символи.

Ограничения върху КСГ с ОП

Условия за прилагане на СА метод на ОП:

- а/ без правила с еднаква дясна част;
- б/ без ϵ -продукции;
- с/ операторна КСГ граматика;
- д/ не повече от едно отношение на ОП (т.е 0 или 1) в коя да е двойка символи, съставена от терминални символи.

ОП управляваща таблица

Отношенията на ОП са възможни между всички терминални символи.

Следва управляващата таблица на ОП да се структурира като квадратна матрица с размер

$$\text{row}(\Sigma) \times \text{row}(\Sigma)$$

ОП управляваща таблица

Да се състави ОП упр таблица за КСграматиката на АИ – пример 1

$$E \rightarrow T \mid E + T$$

$$T \rightarrow F \mid T * F$$

$$F \rightarrow a \mid (E)$$

Управляваща таблица ОП

	(a	*	+)
(x	x	x	x	x
a	x	x	x	x	x
*	x	x	x	x	x
+	x	x	x	x	x
)	x	x	x	x	x

ОП управляваща таблица

$E \rightarrow T \mid E + T$

$LtS(E) = \{ +, *, a, (\}$

$RtS(E) = \{ +, *, a,) \}$

$T \rightarrow F \mid T * F$

$LtS(T) = \{ *, a, (\}$

$RtS(T) = \{ *, a,) \}$

$F \rightarrow a \mid (E)$

$LtS(F) = \{ a, (\}$

$RtS(F) = \{ a,) \}$

ОП релация =

в граматиката на АИ има само една двойка терминали с еднакво предшествие

(=)

ОП управляваща таблица

$E \rightarrow T \mid E + T$

$T \rightarrow F \mid T * F$

$F \rightarrow a \mid (E)$

$LtS(E) = \{ +, *, a, (\}$

$LtS(T) = \{ *, a, (\}$

$LtS(F) = \{ a, (\}$

$RtS(E) = \{ +, *, a,) \}$

$RtS(T) = \{ *, a,) \}$

$RtS(F) = \{ a,) \}$

ОП релация $<$

съставят се продуктивните двойки (terminal, non terminal)

$+, T$

$*, F$

$(, E$

$+ < LtS(T)$

$* < LtS(F)$

$(< LtS(E)$

$+ < \{ *, a, (\}$

$* < \{ a, (\}$

$(< \{ +, *, a, (\}$

ОП управляваща таблица

$E \rightarrow T \mid E + T$

$T \rightarrow F \mid T * F$

$F \rightarrow a \mid (E)$

$LtS(E) = \{ +, *, a, (\}$

$LtS(T) = \{ *, a, (\}$

$LtS(F) = \{ a, (\}$

$RtS(E) = \{ +, *, a,) \}$

$RtS(T) = \{ *, a,) \}$

$RtS(F) = \{ a,) \}$

ОП релация $>$

съставят се продуктивните двойки (non terminal, terminal)

E,+

T,*

E,)

$RtS(E) > +$

$RtS(T) > *$

$RtS(E) >)$

$\{ +, *, a,) \} > +$

$\{ *, a,) \} > *$

$\{ +, *, a,) \} >)$

ОП управляваща таблица

$E \rightarrow T \mid E + T$

$T \rightarrow F \mid T * F$

$F \rightarrow a \mid (E)$

Управляваща таблица ОП

	(a	*	+)
(<	<	<	<	=
a	x	x	>	>	>
*	<	<	>	>	>
+	<	<	<	>	>
)	x	x	>	>	>

Ограничения

Условия за прилагане на СА метод на ОП:

а/ оперторна КСГ граматика;

б/ не повече от едно отношение на ОП (т.е 0 или 1) в коя да е двойка символи, съставена от терминални символи.

Приложение на метод ОП

Приложение на СА ОП метод за конвертиране на АИ от инфиксен запис в обратен полски запис

Дадено: входна лента (АИ в инфиксен запис), ОП управляваща таблица.

Изход: АИ конвертиран в постфиксен (обратен полски) запис.

Приложение на метод ОП

Алгоритъм: входната лента се сканира Л>Д. Проверяват се ОП релациите между символа на върха на стека и текущия входен СИМВОЛ:

```
if ( (top stack символът се предшества (<) от текущия вх с-л) or
      (top stack символът има същото ОП (=) като текущия вх с-л) )
  then { входният символ се записва в стека;
         сканираният вх показалец се инкрементира;
       }
elseif (top stack символът предшества (>) текущия входен символ)
  then { top stack символът се извлича и записва в изх лента ;
         сканираният вх показалец не се инкрементира;
       }
```

Приложение на метод ОП

Демо програма `orprexpr.cpp`.

Съставете ОП упр таблица

$S_p \rightarrow \Delta S \#$

$S \rightarrow a A \mid b B$

$A \rightarrow 0 A 1 \mid 0 1$

$B \rightarrow 0 B 1 \mid 0 1$

$S_p \rightarrow \Delta S \#$

$S \rightarrow a A \mid b B$

$A \rightarrow 0 A 1 \mid c$

$B \rightarrow 0 B 1 \mid 0 1$

Управляваща таблица ОП

	0	1	a	b	c	#
0						
1						
a						
b						
c						
Δ						

Съставете ОП упр таблица

$S \rightarrow \text{if } B \text{ then } S1 \text{ else } S$

$S \rightarrow \text{if } B \text{ then } S$

$S1 \rightarrow \text{if } B \text{ then } S1 \text{ else } S1$

$S \rightarrow c$

$S1 \rightarrow a$

$B \rightarrow B \text{ or } b$

$B \rightarrow b$

Управляваща таблица ОП

	if	then	else	or	a	b	c
--	----	------	------	----	---	---	---

if							
then							
else							
or							
a							
b							
c							

Благодаря
За
Вниманието

5/20/2012

assoc. prof. Stoyan Bonev

34