

Universidade Federal de São Carlos – Departamento de Computação
Construção de Compiladores e Construção de Compiladores 1
Profa. Helena Caseli

Quarta Lista de Exercícios – Análise Sintática Ascendente

1) Construa a tabela de precedência para a gramática abaixo usando o método mecânico. Lembre-se de remover a ambiguidade da gramática caso seja necessário.

$\langle S \rangle ::= (\langle S \rangle + \langle S \rangle) | (\langle S \rangle * \langle S \rangle) | a | b$

R.

A gramática não é ambígua. A existência de pares de “(“ e “)” torna a derivação única para uma dada sentença.

Usando o método mecânico, construção formal:

- $a = b$
 - $(= +, + =), (= *, * =)$
- $a < b$
 - $(< (, (< a, (< b$
 - $+ < (, + < a, + < b$
 - $* < (, * < a, * < b$
- $\$ < b$
 - $\$ < (, \$ < a, \$ < b$
- $a > b$
 - $) >), a >), b >)$
 - $) > +, a > +, b > +$
 - $) > *, a > *, b > *$
- $a > \$$
 - $) > \$, a > \$, b > \$$

Usando o método mecânico, construção informal:

$\text{Primeiros}(S) = \{ (, a, b \}$

$\text{Últimos}(S) = \{ a, b,) \}$

Pares para computar <

- $(\langle S \rangle$
 - $(< \{ (, a, b \}$
- $+ \langle S \rangle$
 - $+ < \{ (, a, b \}$
- $* \langle S \rangle$
 - $* < \{ (, a, b \}$

Pares para computar >

- $\langle S \rangle)$
 - $\{), a, b \} >)$
- $\langle S \rangle +$
 - $\{), a, b \} > +$
- $\langle S \rangle *$
 - $\{), a, b \} > *$

Pares para computar =

- (<S> +
- (= +
- + <S>)
- + =)
- (<S> *
- (= *
- * <S>)
- * =)

$\$ < \text{Primeiros}(S)$

$\$ < \{ (, a, b \}$

$\text{Últimos}(S) > \$$

$\{ \}, a, b \} > \$$

	()	a	b	+	*	\$
(<		<	<	=	=	
)		>			>	>	>
a		>			>	>	>
b		>			>	>	>
+	<	=	<	<			
*	<	=	<	<			
\$	<		<	<			

2) Utilizando a tabela construída no exercício anterior, reconheça a cadeia (**a*b**).

R.

Pilha	Cadeia	Ação
\$	(a*b)\$	empilha
\$(a*b)\$	empilha
\$(a	*b)\$	reduz
\$(S	*b)\$	empilha
\$(S*	b)\$	empilha
\$(S*b)\$	reduz
\$(S*S)\$	empilha
\$(S*S)	\$	reduz
\$S	\$	ACEITA

3) Construa a tabela de precedência para a gramática abaixo:

$\langle S \rangle ::= (\langle L \rangle) \mid a$

$\langle L \rangle ::= \langle L \rangle , \langle S \rangle \mid \langle S \rangle$

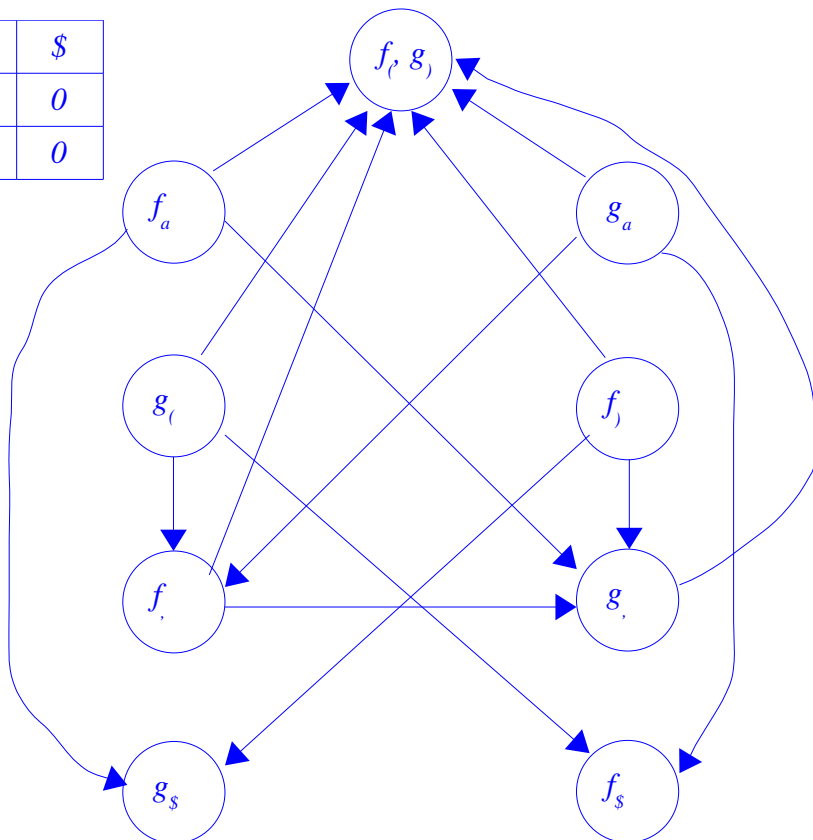
R.

	a	$($	$)$	$,$	$\$$
a			$>$	$>$	$>$
$($	$<$	$<$	$=$	$<$	
$)$			$>$	$>$	$>$
$,$	$<$	$<$	$>$	$>$	
$\$$	$<$	$<$			

4) Encontre as funções de precedência de operadores correspondente à tabela construída no Exercício 3.

R.

	a	$($	$)$	$,$	$\$$
f	2	0	2	2	0
g	3	3	0	1	0



5) Usando as relações de precedência apresentadas na tabela do Exercício 3 ou as funções de precedência geradas no Exercício 4, mostre os passos da análise sintática das sentenças abaixo:

	<i>Pilha</i>	<i>Cadeia</i>	<i>Ação</i>
i. (a, a)	\$	(a,a)\$	<i>empilha</i> ($f(\$) = 0 < g() = 3$)
	\$ (a,a)\$	<i>empilha</i> ($f() = 0 < g(a) = 3$)
	\$ (a	,a)\$	<i>reduz</i> ($f(a) = 2 > g(,) = 1$)
	\$ (S	,a)\$	<i>empilha</i> ($f() = 0 < g(,) = 1$)
	\$ (S ,	a)\$	<i>empilha</i> ($f(,) = 2 < g(a) = 3$)
	\$ (S , a)\$	<i>reduz</i> ($f(a) = 2 > g() = 0$)
	\$ (S , S)\$	<i>reduz</i> ($f(,) = 2 > g() = 0$)
	\$ (L	\$	<i>empilha</i> ($f() = 0 = g(\$) = 0$)
	\$ (L)	\$	<i>reduz</i> ($f() = 2 > g(\$) = 0$)
	\$ S	\$	ACEITA

- ii. (a, (a, a))
- iii. (a, ((a, a), (a,a)))

6) Dada a gramática

$\langle S \rangle ::= \text{if } \langle E \rangle \text{ then } \langle C \rangle \mid \langle C \rangle$
 $\langle E \rangle ::= a$
 $\langle C \rangle ::= b$

a) Construa a tabela SLR.

R. *Passo 1: adicionar a regra $S' \rightarrow S$*

- 0) $\langle S' \rangle ::= \langle S \rangle$
- 1) $\langle S \rangle ::= \text{if } \langle E \rangle \text{ then } \langle C \rangle$
- 2) $\langle S \rangle ::= \langle C \rangle$
- 3) $\langle E \rangle ::= a$
- 4) $\langle C \rangle ::= b$

Passo 2: construir o conjunto de itens

$I_0 = \{S' \rightarrow .S, S \rightarrow .\text{if } E \text{ then } C, S \rightarrow .C, C \rightarrow .b\}$

$t(I_0, S) = \{S' \rightarrow S.\} = I_1$

$t(I_0, \text{if}) = \{S \rightarrow \text{if}.E \text{ then } C, E \rightarrow .a\} = I_2$

$t(I_0, C) = \{S \rightarrow C.\} = I_3$

$t(I_0, b) = \{C \rightarrow b.\} = I_4$

$t(I_2, E) = \{S \rightarrow \text{if } E \text{ then } C.\} = I_5$

$t(I_2, a) = \{E \rightarrow a.\} = I_6$

$t(I_5, \text{then}) = \{S \rightarrow \text{if } E \text{ then } .C, C \rightarrow .b\} = I_7$

$t(I_7, C) = \{S \rightarrow \text{if } E \text{ then } C.\} = I_8$

$t(I_7, b) = \{C \rightarrow b.\} = I_4$

Construir a tabela sintática

$\text{Seguidor}(S') = \{\$ \}$

$\text{Seguidor}(S) = \text{Seguidor}(S') = \{\$ \}$

$\text{Seguidor}(E) = \{\text{then}\}$

$\text{Seguidor}(C) = \text{Seguidor}(S) = \{\$ \}$

Estados	Ações					Transições		
	if	then	a	b	\$	S	E	C
0	s2			s4		1		3
1					OK			
2			s6				5	
3					r2			
4					r4			
5		s7						
6		r3						
7				s4				8
8					r1			

b) Reconheça a cadeia **if a then b** preenchendo os valores da pilha, cadeia e ação a cada passo:

Pilha	Cadeia	Ação
0	if a then b \$	

R.

Pilha	Cadeia	Regra
0	if a then b \$	s2
0 if 2	a then b \$	s6
0 if 2 a 6	then b \$	r3
0 if 2 E 5	then b \$	s7
0 if 2 E 5 then 7	b \$	s4
0 if 2 E 5 then 7 b 4	\$	r4
0 if 2 E 5 then 7 C 8	\$	r1
0 S 1	\$	OK

7) Ordene os três tipos de análise sintática ascendente LR do tipo mais simples e menos poderoso para o mais complexo e mais poderoso, descrevendo brevemente as características de cada um.

R. *Os três tipos de analisadores sintáticos ascendentes LR são:*

- *Simple LR (SLR) – é o mais simples e fácil de aplicar, porém aplicável a uma classe restrita de gramáticas. Tem como base o conjunto de itens LR(0)*
- *Look Ahead LR (LALR) – é o método de poder e complexidade intermediários, funciona para a maioria de linguagens de programação e é a técnica implementada pelo Yacc*
- *LR Canônico – é o mais poderoso, pode ser aplicado a um grande número de LLC. O LR(1), além de considerar os itens LR(0), utiliza também o conjunto de itens LR(1) nos quais há uma marca de verificação à frente muito útil na solução de conflitos.*

8) Dada a gramática

- (1) $\langle E \rangle ::= \langle E \rangle + \langle T \rangle$
- (2) $\langle E \rangle ::= \langle T \rangle$
- (3) $\langle T \rangle ::= \langle T \rangle * \langle F \rangle$
- (4) $\langle T \rangle ::= \langle F \rangle$
- (5) $\langle F \rangle ::= (\langle E \rangle)$
- (6) $\langle F \rangle ::= id$

E a tabela sintática LR

