Universidade Federal de Mato Grosso do Sul
Facom - Faculdade de Computação

Compiladores I - 2023/1
Profa. Bianca de Almeida Dantas

Lista de Exercícios 2 – Análise Sintática Descendente

1. Considere a gramática a seguir:

   $L \rightarrow R\mathbf{a} \mid Q\mathbf{ba}$

   $R \rightarrow \mathbf{aba} \mid \mathbf{caba} \mid R\mathbf{bc}$

   $Q \rightarrow \mathbf{bbc} \mid \mathbf{bc}$

   (a) Por que essa gramática não é LL(1)?

   (b) Transforme essa gramática em uma gramática LL(1).

   (c) Construa os conjuntos FIRST, FOLLOW, e FIRST$^+$ da gramática desenvolvida no item 1b e mostre que ela é, de fato, LL(1).

   (d) Escreva o pseudocódigo dos procedimentos para um analisador sintático descendente recursivo que reconheça a linguagem especificada. Considere a existência das funções `advance()` e `match()`, como vistas em nossas aulas teóricas.

   (e) Construa a tabela de análise sintática descendente não recursiva, com recuperação de erros usando o modo pânico com a função FOLLOW como conjunto de sincronismo (como visto em sala), para a gramática do item 1b.

   (f) Mostre as execuções do analisador sintático descendente não recursivo usando a tabela do item 1e e mostrando o conteúdo da pilha e da entrada e as ações correspondentes, para as cadeias de entrada **cababca** e **baaabcba**.

2. Considere a seguinte gramática:

   $S \rightarrow \mathbf{c}A\mathbf{a}$

   $A \rightarrow \mathbf{c}B \mid B$

   $B \rightarrow \mathbf{bc}B \mid \epsilon$

   (a) Essa gramática é LL(1)? Se não for, obtenha uma gramática LL(1) equivalente.

   (b) Construa os conjuntos FIRST, FOLLOW, e FIRST$^+$ da gramática (após possíveis modificações realizadas no item anterior).

   (c) Construa a tabela de análise sintática descendente não recursiva, sem recuperação de erros, para a gramática utilizada no item anterior.

   (d) Mostre a execução da análise para a entrada **cbca**.

1) $L \to Ra \mid Qba$

$R \to aba \mid caba \mid Rbc$

$Q \to bbc \mid bc$

a) • Ambiguidade em Q.

• $First(R) = Follow(R) = a$

• Recursão à esquerda em R.

b) 1. Eliminando a ambiguidade em Q: (Fatoração à esquerda)

$Q \to bQ'$

$Q' \to bc \mid c$

2. Eliminando a recursão à esquerda em R:

$R \to abaR' \mid cabaR'$

$R' \to bcR' \mid \varepsilon$

Reescrevendo a gramática:

$L \to Ra \mid Qba$

$Q \to bQ'$

$Q' \to bc \mid c$

$R \to abaR' \mid cabaR'$

$R' \to bcR' \mid \varepsilon$

c)

	First	Follow	First +
L	{a, c, b}	{$}	{a, c, b}
R	{a, c}	{a}	{a, c}
R'	{b, ε}	{a}	{b, ε, a}
Q	{b}	{b}	{b}
Q'	{b, c}	{b}	{b, c}

d)

```
void L()
{
      if(token=='a' || token=='c') // FIRST(R)
            {R(); match('a');}
      else // Produção Q
            {Q(); match('b'); match('a');}
}
```

```
void R()
{
      if(token=='a')
      {match('a'); match('b');
       match('a'); R_();}
      else
      {match('c'); match('a');
       match('b'); match('a');
       R_();}
}
```

```
void R_()
{
      if(token=='b')
      {match('b'); match('c'); R_()}
}
```

```
void Q()
{
      match('b'); Q_();
}
```

```
void Q_()
{
      if(token=='b')
      {match('b'); match('c');}
      else
      match('c');
}
```

e)

	a	b	c	$
L	L→Ra	L→Qba	L→Ra	synch
R	R→abaR'		R→cabaR'	
R'	R'→ε	R'→bcR'		
Q		Q→bQ'		
Q'		Q→bc	Q→c	

L → Ra | Qba
Q → bQ'
Q' → bc | c
R → abaR' | cabaR'
R' → bcR' | ε

f)

casado	pilha	entrada	ação
	L$	cababca$	L→Ra
	Ra$	⋮	R→cabaR'
	cabaR'a$	cababca$	casa 'c'
c	abaR'a$	ababca$	casa 'a'
ca	baR'a$	babca$	casa 'b'
cab	aR'a$	abca$	casa 'a'
caba	R'a$	bca$	R'→bcR'
	bcR'a$	bca$	casa 'b'
cabab	cR'a$	ca$	casa 'c'
cababc	R'a$	a$	R'→ε
	a$	a$	casa 'a'
cababca	$	$	casa $
cababca$			

casado	pilha	entrada	ação
	L$	baaabcba$	L→Qba
	Qba$	⋮	Q→bQ'
	bQ'ba$	⋮	casa 'b'
b	Q'ba$	aaabcba$	erro ignora
	Q'ba$	aabcba$	⋮
	⋮	abcba$	⋮
	⋮	bcba$	Q'→bc
	bcba$	bcba$	casa 'b'
bb	cba$	cba$	casa 'c'
bbc	ba$	ba$	casa 'b'
bbcb	a$	a$	casa 'a'
bbcba	$	$	casa $
bbcba	—	—	Fim

② $S \to aAb$

$A \to cB \mid B$

$B \to bcB \mid \varepsilon$

a) É LL(1), pois não tem ambiguidade nem recursão à esquerda.

b)

	First	Follow	First$^+$
S	{a}	{$}	{a}
A	{c, b, ε}	{b}	{c, b, ε}
B	{b, ε}	{b}	{b, ε}

c)

	a	b	c	$
S			S→cAa	
A		A→B	A→cB	
B		B→bcB		

b)

casada	pilha	entrada	ação
	S$	cbca$	S→cAa
	cAa$	cbca$	casa 'c'
c	Aa$	bca$	A→B
	Ba$	⋮	B→bcB
	bcBa$	⋮	casa 'b'
cb	cBa$	ca$	casa 'c'
cbc	Ba$	a$	B→ε
	a$	a$	casa 'a'
cbca	$	$	casa '$'
cbca$	–	–	Fim