

# Análise Sintática

## Aula Prática

Fernando Antônio Asevedo Nóbrega

Instituto de Ciências Matemáticas e de Computação – USP

SCC-206 Introdução à Compilação  
9 de maio de 2012

# Agenda

- 1 Introdução
- 2 Análise Sintática Descendente
  - ASD com Retrocesso
  - ASD Preditiva
- 3 Análise Sintática Ascendente
  - Precedência de Operadores
  - Analisadores LR

# Introdução

## Gramática abordada

$$S \rightarrow (L) \mid a$$
$$L \rightarrow L,S \mid S$$

# Análise Sintática Descendente (ASD)

- Por que é chamada assim?

# Análise Sintática Descendente (ASD)

- Por que é chamada assim?
- Abordagem *top-down*
- Das regras de produção (ñ terminais) se chega aos terminais

# ASD com Retrocesso

Como é realizada?

Reconhecer a cadeia: (a, (a, a))

$S \rightarrow (L) \mid a$

$L \rightarrow L,S \mid S$

# ASD com Retrocesso

Como é realizada?

Reconhecer a cadeia: (a, (a, a))

$$S \rightarrow (L) \mid a$$
$$L \rightarrow L,S \mid S$$

# ASD com Retrocesso

Como é realizada?

Reconhecer a cadeia: (a, (a, a))

$$S \rightarrow (L) \mid a$$
$$L \rightarrow L,S \mid S$$

Isso, não funciona. Por quê?



# ASD com Retrocesso

Como é realizada?

Reconhecer a cadeia: (a, (a, a))

$$S \rightarrow (L) \mid a$$

$$L \rightarrow L,S \mid S$$

Isso, não funciona. Por quê?

Recursividade esquerda

Sem recursividade esquerda

$$S \rightarrow (L) \mid a$$

$$L \rightarrow S,L \mid S$$

# ASD Preditiva

O que a diferencia da ASD com Retrocesso?

$$S \rightarrow (L) \mid a$$
$$L \rightarrow L,S \mid S$$

A gramática é LL(1)?

# ASD Preditiva

O que a diferencia da ASD com Retrocesso?

$$S \rightarrow (L) \mid a$$
$$L \rightarrow L,S \mid S$$

A gramática é LL(1)?

# ASD Preditiva

O que a diferencia da ASD com Retrocesso?

$$S \rightarrow (L) \mid a$$
$$L \rightarrow L,S \mid S$$

A gramática é LL(1)?

$$S \rightarrow (L) \mid a$$
$$L \rightarrow SL'$$
$$L' \rightarrow ,SL' \mid \lambda$$

# ASD Preditiva

O que a diferencia da ASD com Retrocesso?

$$S \rightarrow (L) \mid a$$

$$L \rightarrow L,S \mid S$$

A gramática é LL(1)?

$$S \rightarrow (L) \mid a$$

$$L \rightarrow SL'$$

$$L' \rightarrow ,SL' \mid \lambda$$

E se fosse LL(2), o que mudaria?

# Primeiro e Seguidor

$$S \rightarrow (L) \mid a$$
$$L \rightarrow SL'$$
$$L' \rightarrow \lambda \mid ,SL'$$

# Primeiro e Seguidor

$$S \rightarrow (L) \mid a$$

$$L \rightarrow SL'$$

$$L' \rightarrow ,SL' \mid \lambda$$

## Conjunto Primeiro

$$P(S) = \{ ( a \}$$

$$P(L) = P(S) = \{ ( a \}$$

$$P(L') = \{ , \lambda \}$$

## Conjunto Seguidor

$$S(S) = \{ \$ \} + P(L') + S(L')^a = \{ \$ , \lambda ) \}$$

$$S(L) = \{ ) \}$$

$$S(L') = S(L) + S(L') = \{ ) \}$$


---

<sup>a</sup>Pois  $L'$  pode gerar  $\lambda$

# Tabela Sintática



# Tabela Sintática

	(	)	a	\$	,	$\lambda$
<b>S</b>	$S \rightarrow (L)$		$S \rightarrow a$			
<b>L</b>	$L \rightarrow SL'$		$L \rightarrow SL'$			
<b>L'</b>		$L' \rightarrow \lambda$			$L' \rightarrow ,SL'$	$L' \rightarrow \lambda$

# Análise de Precedência de Operadores

Mais usual para gramática de expressões matemáticas

## Tipo de Gramática

- Não há símbolos não terminais adjacentes
- Não há produções que derivam a cadeia nula

# Análise de Precedência de Operadores

Vamos usar essa gramática?

$$S \rightarrow (L) \mid a$$
$$L \rightarrow SL'$$
$$L' \rightarrow ,SL' \mid \lambda$$

Gramática correta

$$S \rightarrow (L) \mid a$$
$$L \rightarrow L,S \mid S$$

# Análise de Precedência de Operadores

Vamos usar essa gramática?

$$S \rightarrow (L) \mid a$$

$$L \rightarrow SL'$$

$$L' \rightarrow ,SL' \mid \lambda$$

Não, por quê?

Gramática correta

$$S \rightarrow (L) \mid a$$

$$L \rightarrow L,S \mid S$$

# Tabela de Precedência de Operadores

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

# Analísadores LR

- Mais poderosos
- Analísadores em formato de autômatos
- Por que são chamados assim?
  - Da esquerda para a direita, derivação mais a direita

# Analísadores LR

- Mais poderosos
- Analísadores em formato de autómatos
- Por que são chamados assim?
  - Da esquerda para a direita, derivação mais a direita

# Simple LR (SLR)

Funciona com base no uso de uma tabela SLR

- A tabela SLR é construída pelo conjunto canônico LR(0)
  - Algoritmo do ponto

Construir o conjunto canônico para:

$S \rightarrow (L) \mid a$

$L \rightarrow L, S$

$L' \rightarrow S$



# Simple LR (SLR)

Funciona com base no uso de uma tabela SLR

- A tabela SLR é construída pelo conjunto canônico LR(0)
  - Algoritmo do ponto

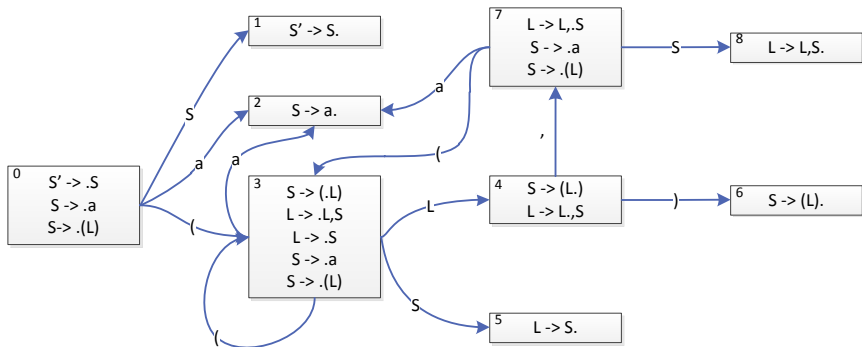
Construir o conjunto canônico para:

$S \rightarrow (L) \mid a$

$L \rightarrow L, S$

$L' \rightarrow S$

# Conjunto Canônico



# Algoritmo da Tabela SLR

- 1 Cria-se uma regra auxiliar que gera o estado inicial
- 2 Enumera-se as regras de produção, inclusive a regra auxiliar
- 3 Calcula-se os seguidores para todo não terminal
- 4 Calcula-se o conjunto fechamento para todas as regras (Algoritmo do ponto)
- 5 Preencher a tabela (tem-se que  $n(\beta)$  é a numeração da regra):
  - Se  $I_i$  com um terminal  $\alpha$  gera um estado  $I_j$ , preenche-se com  $sj$  a linha  $i$  coluna  $\alpha$
  - Se  $I_i$  com uma regra  $\beta$  gera um novo estado  $I_j$ , preenche com  $j$  a linha  $i$  coluna  $\beta$
  - Se  $I_i$  consome uma regra  $\beta$  (o ponto chega ao final), preenche-se com  $rn(\beta)$  na linha  $i$  e colunas dos seguidores de  $\beta$

## Tabela SLR

Estados	a	(	)	,	\$	S	L
0	s2	s3				1	
1					OK		
2			r1	r2	r3		
3	s2	s3				5	4
4			s6	s7			
5			r4	r4			
6			r2	r2	r2		
7	s2	s3				8	
8			r3	r3			