

Compiladores – Tabelas ACTION e GOTO

Fabio Mascarenhas – 2018.1

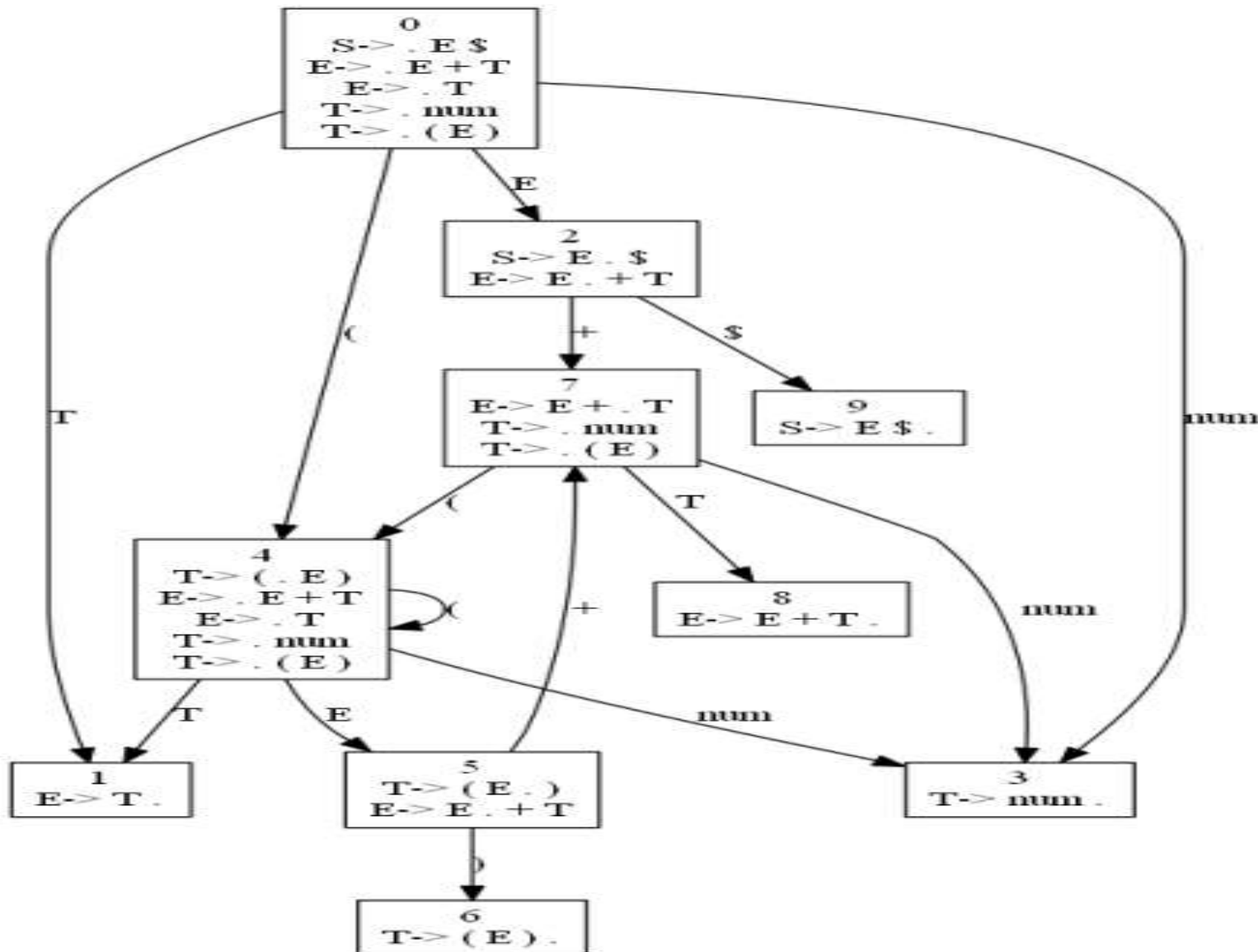
<http://www.dcc.ufrj.br/~fabiom/comp>

Otimizando o analisador SLR

- A implementação do analisador SLR não precisa executar o autômato em toda a pilha sempre
- Podemos associar um número de estado a cada elemento da pilha (com outra pilha, por exemplo), para ser o estado onde o autômato se encontra quando percorreu a pilha até aquele elemento
- Um shift empilha o estado resultante de fazer a transição do estado que estava no topo da pilha antes do shift
- Um reduce empilha o estado resultante de fazer a transição do estado que estava no topo da pilha depois de desempilhar o lado direito

Estados na pilha

$S \rightarrow E \$$
 $E \rightarrow E + T$
 $E \rightarrow T$
 $T \rightarrow \text{num}$
 $T \rightarrow (E)$



Analizando num + (num + num) \$

$\emptyset \mid \text{num} + (\text{num} + \text{num}) \$$
 $\emptyset \text{ num } 3 \mid + (\text{num} + \text{num}) \$$
 $\emptyset T 2 \mid + (\text{num} + \text{num}) \$$
 $\emptyset E 2 \mid + (\text{num} + \text{num}) \$$
 $\emptyset E 2 + 7 \mid (\text{num} + \text{num}) \$$
 $\emptyset E 2 + 7 (4 \mid \text{num} + \text{num}) \$$
 $\emptyset E 2 + 7 (4 \text{ num } 3 \mid + \text{num}) \$$
 $\emptyset E 2 + 7 (4 T 2 \mid + \text{num}) \$$

$\emptyset E 2 + 7 (4 E 5 \mid + \text{num}) \$$
 $\emptyset E 2 + 7 (4 E 5 + 7 \mid \text{num}) \$$
 $\emptyset E 2 + 7 (4 E 5 + 7 \text{ num } 3 \mid) \$$
 $\emptyset E 2 + 7 (4 E 5 + 7 T 8 \mid) \$$
 $\emptyset E 2 + 7 (4 E 5 \mid) \$$
 $\emptyset E 2 + 7 (4 E 5) 6 \mid \$$
 $\emptyset E 2 + 7 T 8 \mid \$$
 $\emptyset E 2 \mid \$$
 $\emptyset E 2 \mid 9 \mid \rightarrow \emptyset 5 \mid \checkmark$

S \rightarrow E \$
 E \rightarrow E + T
 E \rightarrow T
 T \rightarrow num
 T \rightarrow (E)

Tabelas ACTION e GOTO

- Podemos construir uma grande tabela a partir do autômato, e guiar o analisador a partir dessa tabela
- As linhas são estados, as colunas símbolos (terminais e não-terminais)
- A parte da tabela dos terminais se chama ACTION
consulta c/ o lookahead
 - Ela diz o que o autômato deve fazer se o próximo token for o terminal
- A parte dos não-terminais se chama GOTO
 - Ela diz para qual estado ir após uma redução para aquele não-terminal

Preenchendo a tabela SLR

- Para cada estado:
 - Transições em terminais viram entradas S_n para aquele terminal, onde n é o estado de destino (ACTION)
 - Transições em não-terminais viram entradas n para aquele não-terminal (GOTO)
 - Itens de redução viram entradas R_n para todos os terminais no FOLLOW do não-terminal da regra, onde n é o número de regra (ACTION)
 - Itens de redução para o símbolo inicial da gramática e o final da entrada geram entradas A , para *accept* (ACTION)

Tabelas ACTION e GOTO

	\$	+	m	()	EOF	S	E	T
0			S3 S4					2	1
1	RC	RC			KC				
2	S9	S7							
3	RD	RD			KD				
4			S3 S4				5	1	
5		S7			S6				
6	RE	RE			KE				
7			S3 S4						8
8	RB	RB			KB				
9						A			

- A S -> E \$
- B E -> E + T
- C E -> T
- D T -> num
- E T -> (E)

Analísadores LR de tabela

- Buracos na tabela indicam erros sintáticos
- Tentar adicionar uma entrada em uma célula já preenchida é um conflito, usar as regras para resolução
- Todos os métodos LR com um token de lookahead usam a mesma estrutura de tabela, o que varia é só o método de preenchimento, e o tamanho da tabela no caso da análise LR(1)
- As tabelas para analisadores LR(0), SLR e LALR de uma dada gramática têm o mesmo tamanho

Trecho da tabela de TINY

$E \rightarrow E + E$	\rightarrow	11
$E \rightarrow E \cdot + E$	\rightarrow	12
$E \rightarrow E \cdot * E$	\rightarrow	13
$E \rightarrow E \cdot - E$	\rightarrow	14
$E \rightarrow E \cdot / E$	\rightarrow	15
$E \rightarrow E \cdot < E$	\rightarrow	16
$E \rightarrow E \cdot = E$	\rightarrow	16

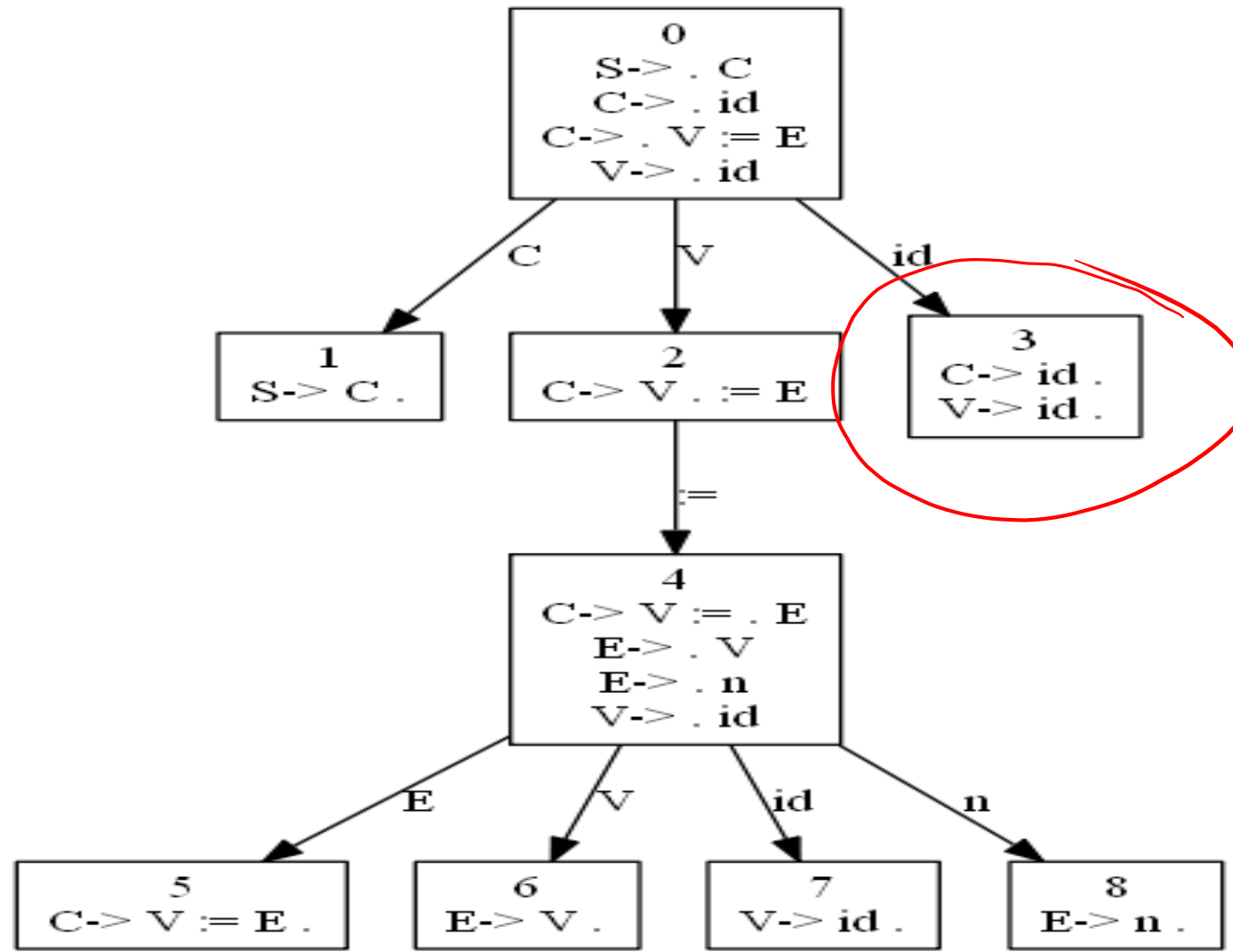
$Follow(E) = \{ +, -, *, /, <, =,), \text{then}, \text{end}, \text{unl}, \text{eof} \}$

	$+$	$-$	$<$	$=$	$*$	$/$	then end unl eof)	...			
10	R	R	R	R	S13	S14	R	R	R	R	R
	conflictos resolvidos										

Limitações do método SLR

- Existem gramáticas que não são SLR:

$S \rightarrow C$
 $C \rightarrow id$
 $C \rightarrow V := E$
 $V \rightarrow id$
 $E \rightarrow V$
 $E \rightarrow n$



$Follow(C) = \{eof\}$
 $Follow(V) = \{eof, :=\}$

Limitações do método SLR

- Existem métodos de análise mais poderosos
- LALR associa um conjunto similar ao FOLLOW para cada item, mas mais preciso que o FOLLOW
- LR(1) e LR(k) mudam o conceito de item, gerando um autômato maior e mais preciso