



Análise Sintática Ascendente

Tabelas LR(0) e SLR(1)

GCC130 - Compiladores

Tabelas LR

- Os diferentes parsers LR tem a mesma estrutura
- O que varia é a Tabela de Parsing, que pode ser:
 - LR(0)
 - SLR(1)
 - CLR(1)
 - LALR(1)

Tabelas LR

- As tabelas LR são do seguinte formato

Ações a serem realizadas (Shift ou Reduce) conforme o símbolo lido na entrada e o estado atual do Parser.

Desvios para o próximo estado, após operações de Reduce.

Estados, configurações ou conjuntos de itens LR. Simulam estados de um autômato.

	ACTION			GO TO
	a	b	\$	S
0	S2			1
<u>1</u>			<u>ACC</u>	
2	S2	S4		3
3		S5		
4	R2	R2	R2	
5	R1	R1	R1	

Construção da Tabela LR(0)

- Considere a gramática

$S \rightarrow aSb$

$S \rightarrow ab$

- Passo 1: Estender a gramática (criar novo estado inicial que aponta para o estado inicial original)

(0) $S' \rightarrow S$

(1) $S \rightarrow aSb$

(2) $S \rightarrow ab$

Construção da Tabela LR(0)

- Passo 2: Calcular os conjuntos de ITENS LR(0)
- Configuração ou ítem (ou ítem LR(0)) = uma regra da gramática com um ‘ponto’ em algum lugar à direita.
- Ex:
 - $A \rightarrow \bullet ab$ (significa que está aguardando a leitura de “a”)
 - $A \rightarrow a\bullet b$ (significa que já fez a leitura de “a” e aguarda a leitura de “b”)
 - $A \rightarrow ab\bullet$ (significa que já fez a leitura de “ab” e está pronto para a operação REDUCE($A \rightarrow ab$))

Construção da Tabela LR(0)

- Passo 2: Calcular os conjuntos de ITENS LR(0)
 - Calcula-se os conjuntos canônicos de itens= todos os itens alcançáveis a partir de um conjunto de regras da gramática.
 - Cada conjunto será um estado do parser
- O conjunto inicial é dado pela produção inicial (estendida), conforme segue:

$$S' \rightarrow \bullet S$$
- Em seguida aplica-se a propriedade fechamento:
- Propriedade de Fechamento:
 - Se $T \rightarrow X_1 \dots X_i \bullet X_{i+1} \dots X_n$ está em um conjunto, e X_{i+1} é um não-terminal que deriva em α , então também está no conjunto: $X_{i+1} \rightarrow \dots \rightarrow \bullet \dots \alpha$

Construção da Tabela LR(0)

- Em seguida aplica-se a propriedade fechamento:
- Propriedade de Fechamento:
 - Se $T \rightarrow X_1 \dots X_i \bullet X_{i+1} \dots X_n$ está em um conjunto, e X_{i+1} é um não-terminal que deriva em α , então também está no conjunto:
$$X_{i+1} \rightarrow \bullet \alpha$$
- Ou seja, sempre que o ponto precede um não-terminal X , devem ser incluídas no conjunto todas as produções em que X está do lado esquerdo.

Construção da Tabela LR(0)

- Ex:

Estado 0

$S' \rightarrow \bullet S$
 $S \rightarrow \bullet aSb$
 $S \rightarrow \bullet ab$

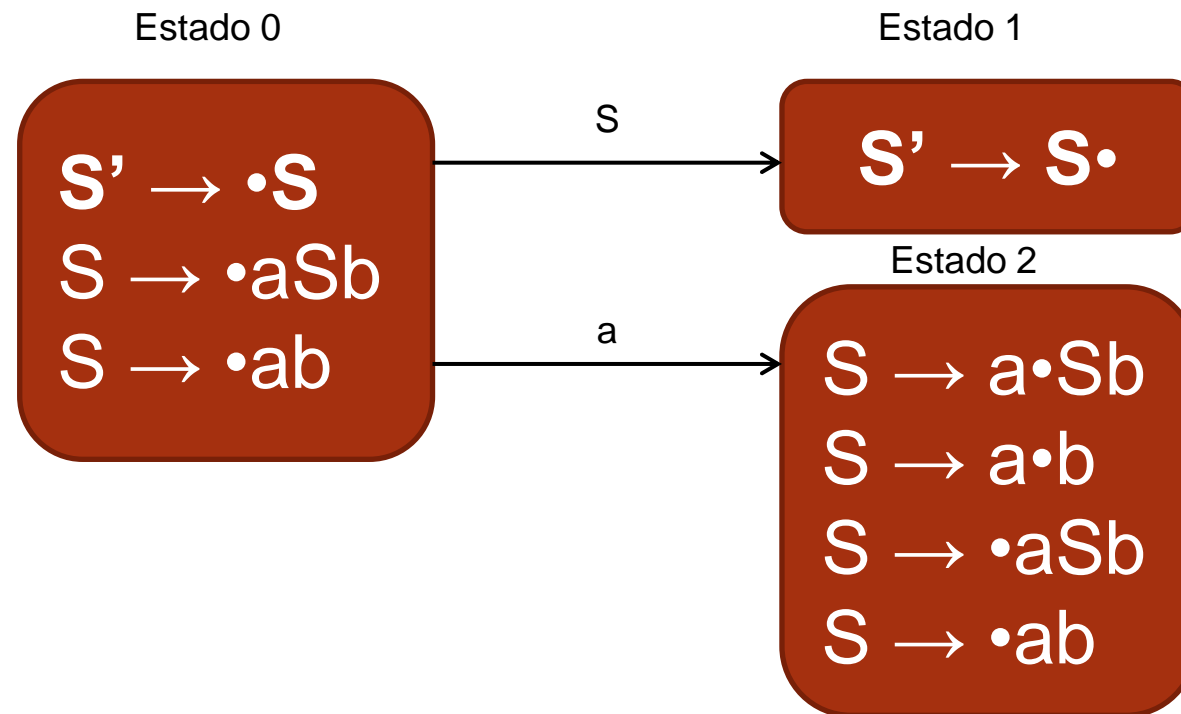
A partir do item $S' \rightarrow \bullet S$, a propriedade de fechamento diz que todas as produções em que S está do lado esquerdo devem ser adicionadas ao conjunto.

Construção da Tabela LR(0)

- Passo 3: Montar o autômato
 - Para cada item em que há um ponto precedendo um símbolo “a” (ou seja, em que está aguardando uma leitura) cria-se uma transição para um novo estado.
 - A aresta é rotulada pelo símbolo a ser lido (“a”), e o novo estado inclui os itens LR representando a situação em que o símbolo “a” já foi lido (i.e., o ponto move para a direita).
 - Repete-se o passo 2 sobre o novo estado (aplicando-se a propriedade de fechamento)

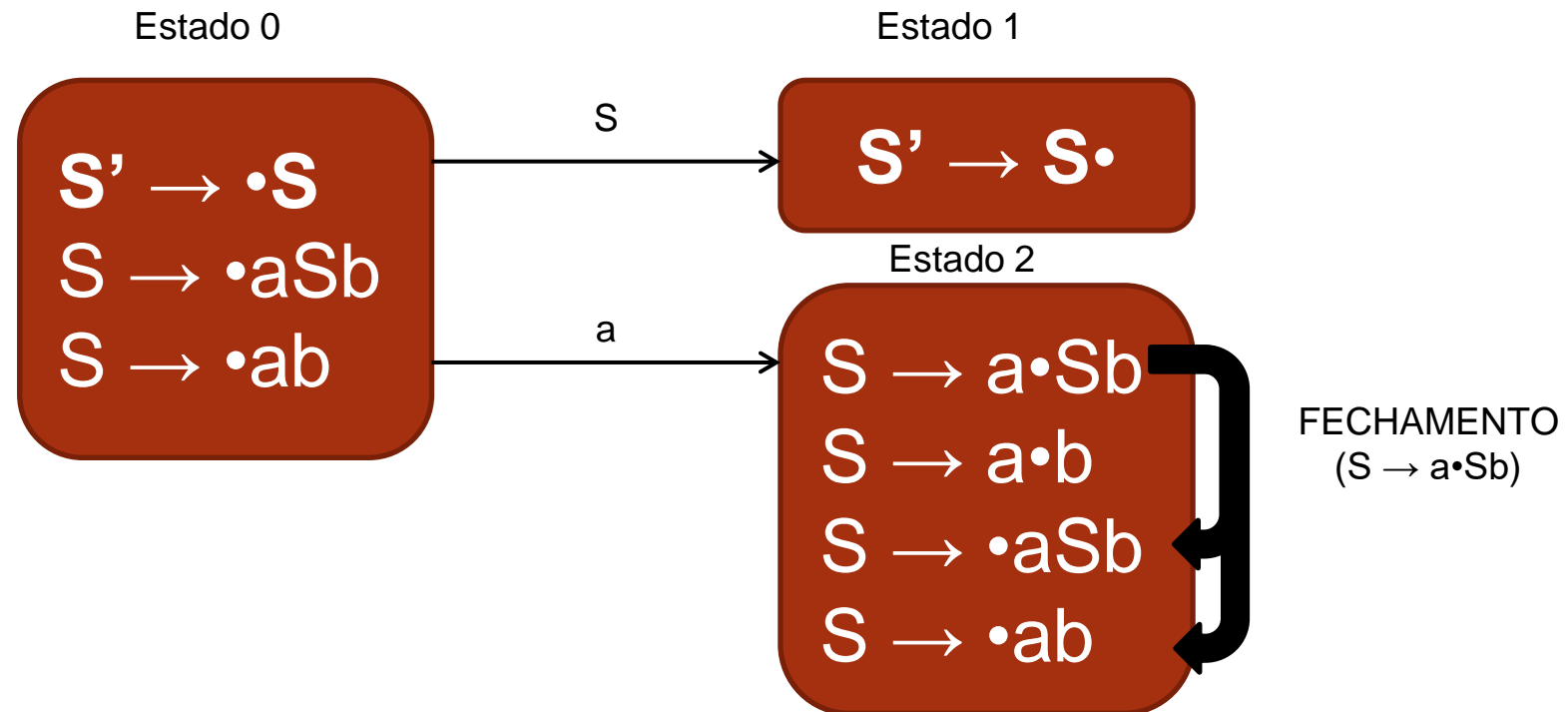
Construção da Tabela LR(0)

• Ex:



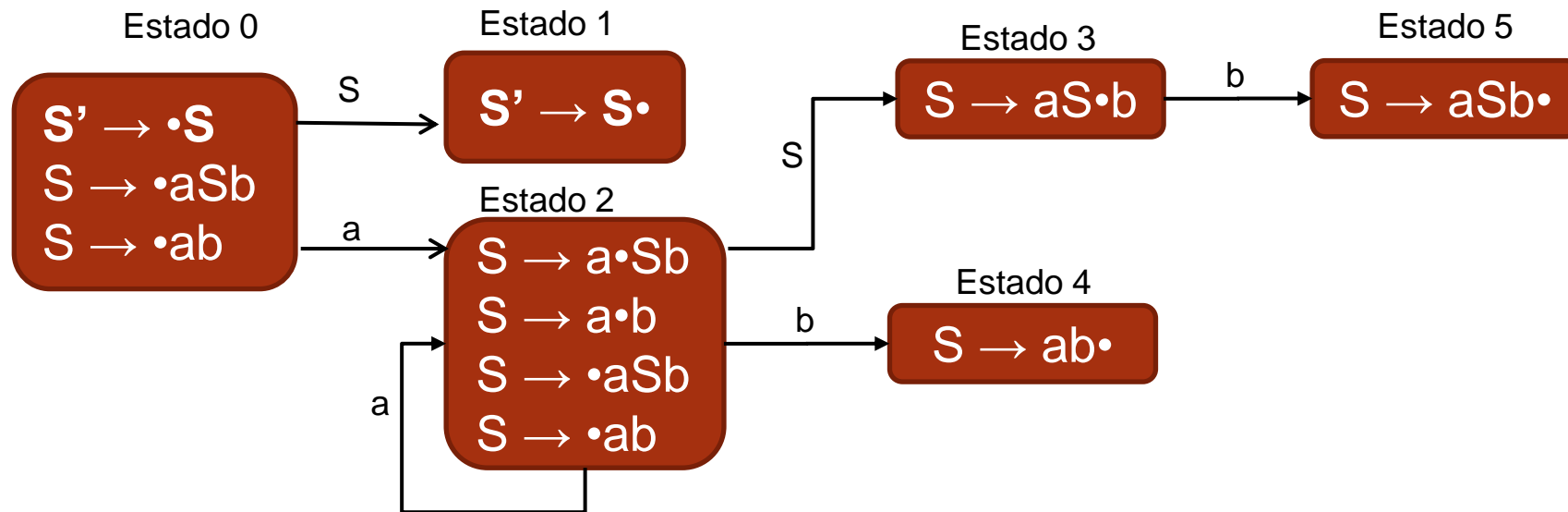
Construção da Tabela LR(0)

• Ex:



Construção da Tabela LR(0)

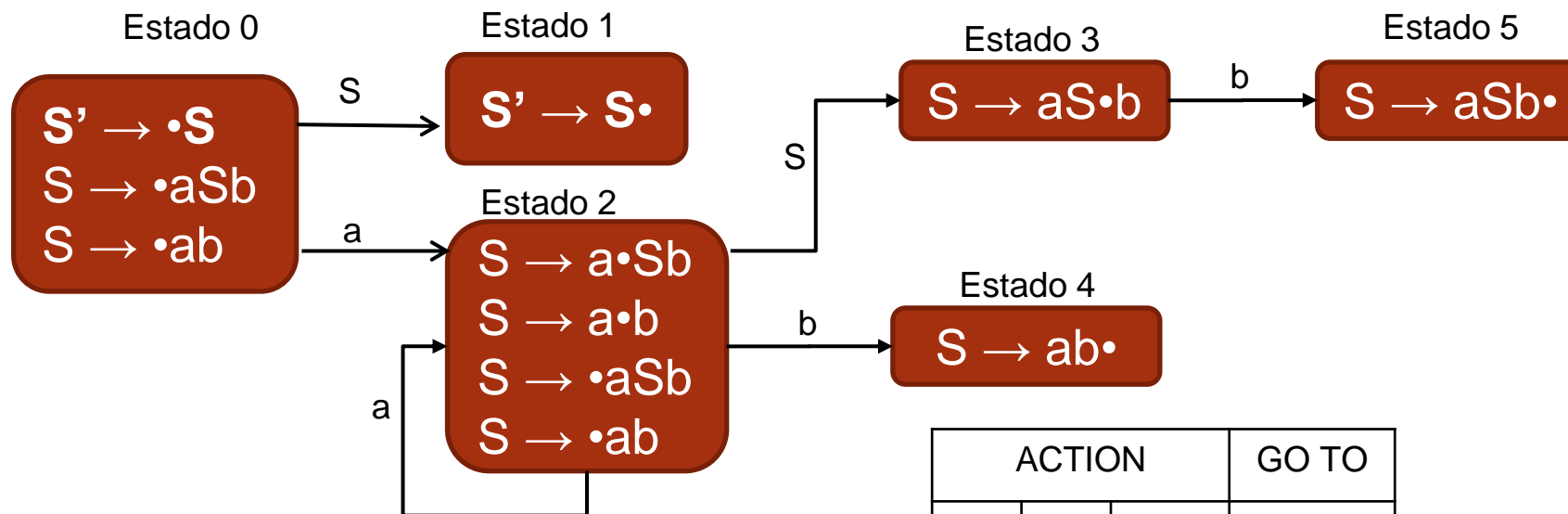
• Ex:



Construção da Tabela LR(0)

- Construir a tabela, onde
 - Há 1 linha para cada estado do autômato.
 - Duas seções de colunas:
 - ACTION: uma coluna para cada terminal mais o \$;
 - GOTO: uma coluna para cada não-terminal.
- Operações de SHIFT e GOTO
- Para cada estado I , para cada aresta de saída com símbolo “ a ”
 - Se “ a ” é terminal, adicionar operação de shift ($S<\text{estado alvo}>$) na célula $[I,a]$ (seção ACTION)
 - Se “ a ” é não terminal, adicionar o estado destino na célula $[I,a]$ (seção GOTO)

Construção da Tabela LR(0)

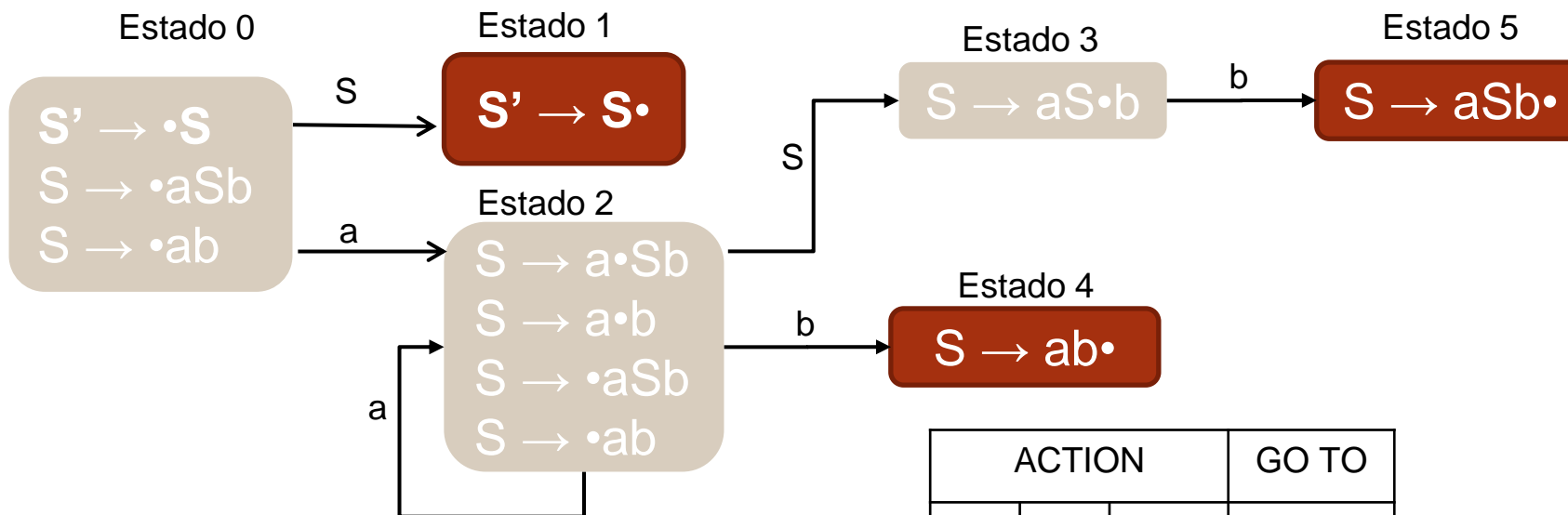


	ACTION			GO TO
	a	b	\$	
0	S2			1
1				
2	S2	S4		3
3		S5		
4				
5				

Construção da Tabela LR(0)

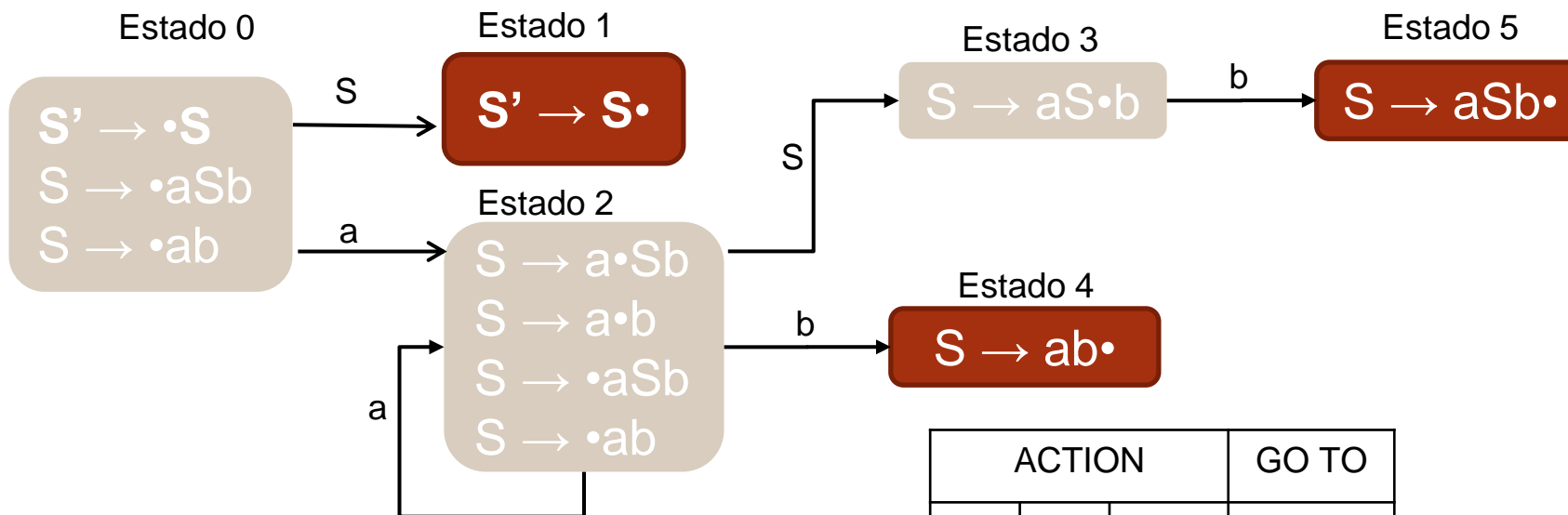
- Operações de REDUCE e ACCEPT
- Para cada estado I que contenha itens do tipo
$$A \rightarrow \alpha \bullet$$
 - Se $A \rightarrow \alpha \bullet$ é a nova produção inicial da gramática estendida, então adicione a ação ACCEPT na célula $[I, \$]$
 - Senão, adicione a operação de REDUCE($A \rightarrow \alpha \bullet$) em todas as células da linha I , na seção ACTION da tabela

Construção da Tabela LR(0)



	ACTION			GO TO
	a	b	\$	
0	S2			1
1			ACC	
2	S2	S4		3
3		S5		
4	R2	R2	R2	
5	R1	R1	R1	

Construção da Tabela LR(0)



	ACTION			GO TO
	a	b	\$	
0	S2			1
1			ACC	
2	S2	S4		3
3		S5		
4	R2	R2	R2	
5	R1	R1	R1	

Conflitos

- Durante a montagem da tabela, podem ser identificados conflitos, ou seja, células em que exista mais de uma entrada.
 - Casos de não-determinismo
- Caso haja conflito, significa que a gramática não é LR(0), portanto as palavras geradas por essa gramática não podem ser analisadas por um parser LR(0).

Conflitos

- (1) $E \rightarrow T + E$
- (2) $E \rightarrow T$
- (3) $T \rightarrow n$

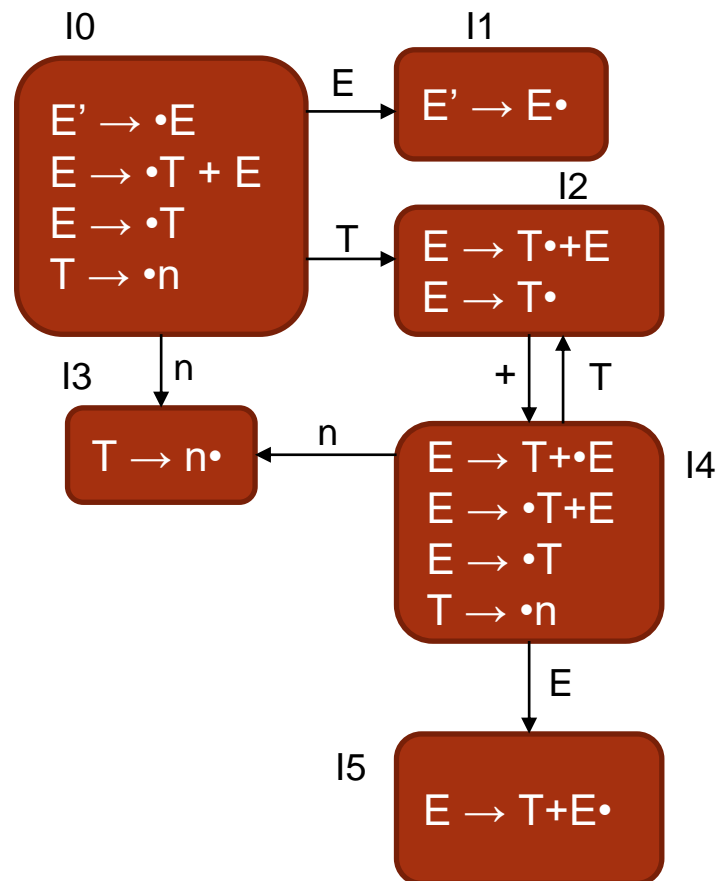


Tabela LR(0)

	ACTION			GO TO	
	+	n	\$	E	T
0		S3		1	2
1			ACC		
2	S4 R2	R2	R2		
3	R3	R3	R3		
4		S3		5	2
5	R1	R1	R1		

Conflitos

- Há dois possíveis tipos de conflito:
 - Conflitos Shift-Reduce (SR)
 - Conflitos Reduce-Reduce (RR)

$S \rightarrow \cdot b$
 $S \rightarrow a \cdot$

ACTION				GO TO
	a	b	\$	S
0	R	<u>S/R</u>	R	

$S \rightarrow b \cdot$
 $S \rightarrow a \cdot$

ACTION				GO TO
	a	b	\$	S
0	<u>R/R</u>	<u>R/R</u>	<u>R/R</u>	

Tabela SLR(1)

- Mesmos procedimentos utilizados na montagem da Tabela LR(0).
- No entanto, o preenchimento de células referentes a operações de REDUCE é diferente
- A ideia é limitar as decisões de redução aos casos que o terminal que segue na entrada é compatível com a redução.
 - Look-ahead
- Seja um item $A \rightarrow a\bullet$, em determinado estado I , só serão adicionadas ações REDUCE($A \rightarrow a$) nas colunas referentes aos símbolos que pertencem ao conjunto SEGUIDORES(A).

Conflitos

(1) $E \rightarrow T + E$

(2) $E \rightarrow T$

(3) $T \rightarrow n$

$\text{SEGUIDORES}(E) = \{\$, +\}$

$\text{SEGUIDORES}(T) = \{\$, +\}$

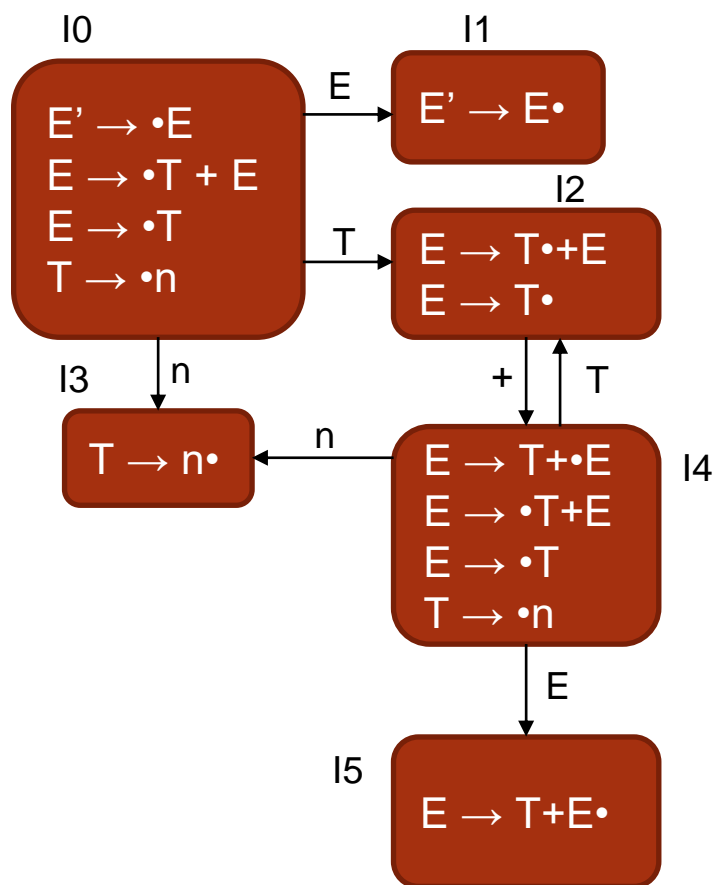


Tabela SLR(1)

	ACTION			GO TO	
	+	n	\$	E	T
0		S3		1	2
1			ACC		
2	S4		R2		
3	R3		R3		
4		S3		5	2
5			R1		

Conflitos na Tabela SLR(1)

Shift-Reduce (SR)

$B \rightarrow \cdot b$
 $A \rightarrow a \cdot$

Somente se
“b” pertence a $\text{SEGUIDORES}(A)$

Reduce-Reduce(RR)

$B \rightarrow b \cdot$
 $A \rightarrow a \cdot$

Somente se
 $\text{SEGUIDORES}(A) \cap \text{SEGUIDORES}(B) \neq \emptyset$

Se ainda assim, houver conflito, a gramática não é SLR(1)



GG

Referência bibliográfica

- Aho, A. V., Lam, M. S., Sethi, R. e Ullman, J. D. (2008).
Compiladores: princípios, técnicas e ferramentas, 2.ed. São Paulo. Pearson Addison Wesley.
 - Capítulo 4 – Análise Sintática