

Compiladores e Computabilidade

Prof. Leandro C. Fernandes
UNIP – Universidade Paulista, 2013

The background of the slide is a blurred image of PHP code. Visible snippets include variable declarations like \$host, \$username, \$password, \$database, and \$charset, and a function definition for mysql_connect. The code is color-coded with green for keywords and blue for variables.

(2ª fase da Análise)

ANÁLISE SINTÁTICA ASDESCENDENTE

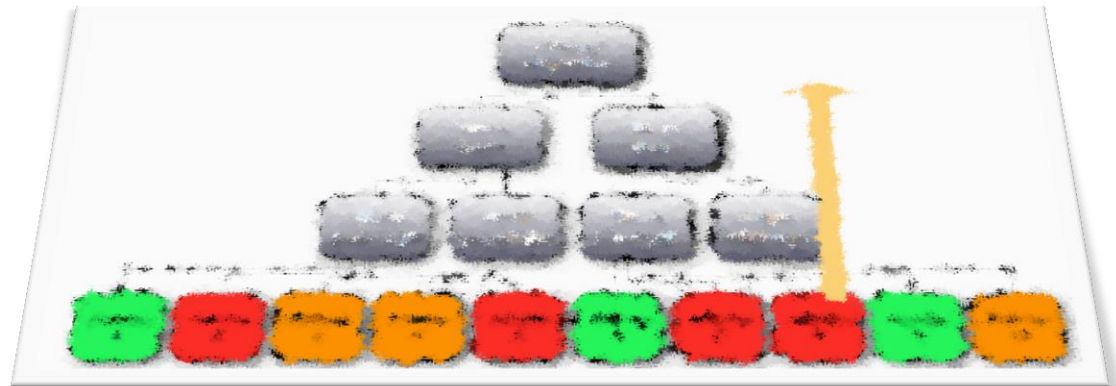
Tipos de Analisadores Sintáticos

Métodos Ascendentes (*Bottom-up*):

- Constroem a árvore sintática de baixo para cima (das folhas para a raiz), ou seja, reduz os símbolos da sentença até alcançar o símbolo inicial da gramática.
 - Analisadores LR
 - Analisadores LALR

A análise sintática ...

- **Tarefa:** Dada uma gramática livre de contexto G e uma sentença s , o analisador sintático deve verificar se s pertence a linguagem gerada por G .
 - O analisador tenta construir a árvore de derivação para s segundo as regras de produção dadas pela gramática G .
 - Se esta tarefa for possível, o programa é considerado sintaticamente correto.



Análise Ascendente

ANÁLISE LR(1)

Análise Ascendente

- A árvore de derivação correspondente a x é construída de *baixo para cima*, ou seja, das folhas, onde se encontra x , para a raiz (o símbolo inicial S).
- Nos métodos ascendentes (*bottom-up*), temos de decidir qual a regra $A \rightarrow \beta$ a ser aplicada e devemos encontrar nós vizinhos rotulados com os símbolos de β . A *redução* pela regra $A \rightarrow \beta$ consiste em acrescentar à árvore um nó A , cujos filhos são os nós correspondentes aos símbolos de β .

Exemplo

- Suponha a gramática: Considere a cadeia: $a+a*a$

(1) $E \rightarrow E + T$

(2) $E \rightarrow T$

(3) $T \rightarrow T * F$

(4) $T \rightarrow F$

(5) $F \rightarrow (E)$

(6) $F \rightarrow a$

Podemos obtê-la:

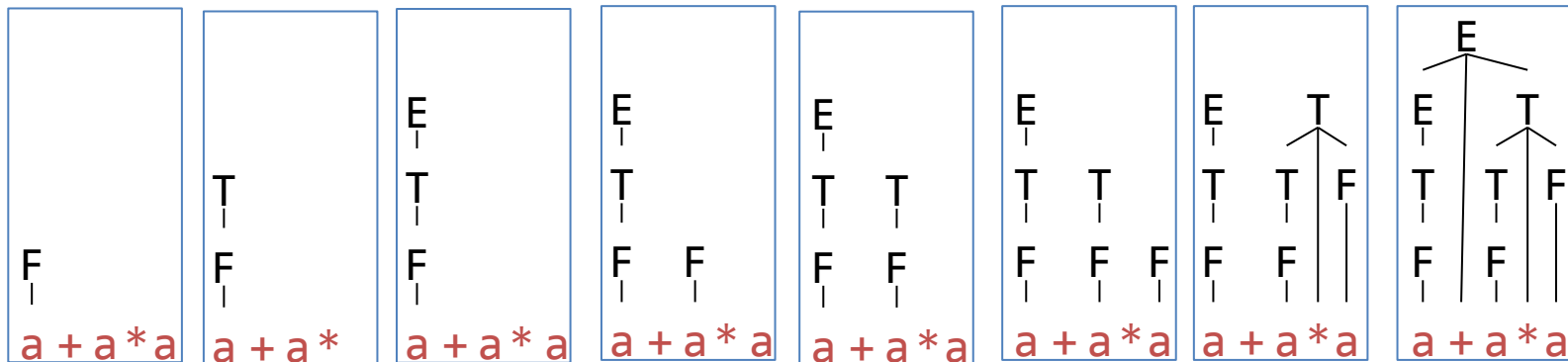
$$\begin{aligned} E &\Rightarrow E+T \Rightarrow E+T*F \Rightarrow E+T*a \Rightarrow \\ &E+F*a \Rightarrow E+a*a \Rightarrow T+a*a \Rightarrow \\ &F+a*a \Rightarrow a+a*a \end{aligned}$$

Derivação *right-most*.

Exemplo

Construindo a árvore de derivação para a cadeia, devemos tomar como base a derivação *right-most* invertida:

$$a+a*a \Leftarrow F+a*a \Leftarrow T+a*a \Leftarrow E+a*a \Leftarrow E+F*a \Leftarrow E+T*a \Leftarrow E+T*F \Leftarrow E+T \Leftarrow E$$



Análise Ascendente

- A representação do processo será feita através de uma pilha de dados
- Para tal, temos:
 - configurações (a, y) ,
 - a = o conteúdo da pilha
 - y = o *resto* da entrada ainda não analisada.
- Existem duas formas de transição:
 - *redução pela regra* $A \rightarrow \beta$:
permite passar da configuração $(\alpha\beta, y)$ para $(\alpha A, y)$.
 - *empilhamento, ou deslocamento de um terminal* a :
permite passar da configuração (α, ay) para $(\alpha a, y)$.

Exemplo

Pilha	(resto da) entrada	derivação esquerda
	a+a*a	a+a*a
a	+a*a	
F	+a*a	$\Leftarrow F+a*a$
T	+a*a	$\Leftarrow T+a*a$
E	+a*a	$\Leftarrow E+a*a$
E+	a*a	
E+a	*a	
E+F	*a	$\Leftarrow E+F*a$
E+T	*a	$\Leftarrow E+T*a$
E+T*	a	
E+T*a	ϵ	
E+T*F	ϵ	$\Leftarrow E+T*F$
E+T	ϵ	$\Leftarrow E+T$
E	ϵ	$\Leftarrow E$

Gramática sLR(1)

O nome sLR(1) indica que:

- A variante mais *simples* dos métodos LR(1)
- A cadeia é lida da esquerda para a direita (L=*left-to-right*);
- O analisador constroi uma derivação direita (R=*rightmost*) invertida;
- Apenas 1 símbolo do resto da entrada é examinado.

Gramática Aumentada

- Para simplificar a identificação do término do processo de análise, acrescentamos à gramática uma nova regra inicial $S' \rightarrow S$, sendo S o símbolo inicial original. S' é um símbolo novo, que passa a ser o símbolo inicial da *gramática aumentada*.
 - Essa regra recebe o número 0. Assim, uma redução pela regra 0 indica o fim da análise, já que S' nunca aparece à direita nas regras da gramática.
- A gramática aumentada é usada na construção do analisador sLR(1) da gramática original.

Entendendo a notação ...

- A construção deste analisador se baseia em itens. Um *item* $A \rightarrow \alpha \bullet \beta$ indica a possibilidade de que, no ponto atual em que se encontra a análise,
 - a regra $A \rightarrow \alpha \beta$ foi usada na derivação da cadeia de entrada;
 - os símbolos terminais derivados de α já foram encontrados;
 - falta encontrar os símbolos terminais derivados de β .
- Desta forma o ponto (\bullet) indica o progresso da análise.

Entendendo a notação ...

- Por exemplo:

$A \rightarrow \bullet \gamma$ indica o início da busca por (uma cadeia derivada de) um γ , enquanto

$A \rightarrow \gamma \bullet$ indica o fim da busca por um γ , ou seja, o momento em que a redução de γ para A pode ser executada.

O estado do processo de análise

- Num dado momento, várias possibilidades precisam ser consideradas, e, por essa razão, representamos um *estado* do processo de análise por um *conjunto de itens*.
- O estado inicial do processo de análise tem um item inicial $S' \rightarrow \bullet S$, proveniente da regra inicial, e pode ser entendido como "só falta encontrar (uma cadeia derivada de) um S'' ".

O estado do processo de análise

- De acordo com as regras da gramática, é necessário acrescentar a cada estado as possibilidades correspondentes.
- Assim, quando um estado contém um item $A \rightarrow \alpha \bullet B \beta$, itens correspondentes às regras de B devem ser acrescentados, para dirigir a busca por B. Esses são os itens da forma $B \rightarrow \bullet \gamma$, para todas as regras $B \rightarrow \gamma$ de B. Esse processo, repetido enquanto for necessário, é denominado o *fechamento* do estado. Assim, o estado inicial é o fechamento de $\{ S' \rightarrow \bullet S \}$.

Exemplo

- (0) $S' \rightarrow E$
- (1) $E \rightarrow E + T$
- (2) $E \rightarrow T$
- (3) $T \rightarrow T * F$
- (4) $T \rightarrow F$
- (5) $F \rightarrow (E)$
- (6) $F \rightarrow a$

- Os itens da gramática aumentada para esse exemplo, serão 20:

$S' \rightarrow \bullet E, S' \rightarrow E \bullet,$

$E \rightarrow \bullet E + T, E \rightarrow E \bullet + T, E \rightarrow E + \bullet T,$

$E \rightarrow E + T \bullet, \dots$

\dots

$F \rightarrow \bullet a, F \rightarrow a \bullet$

Exemplo

- (0) $S' \rightarrow E$
- (1) $E \rightarrow E + T$
- (2) $E \rightarrow T$
- (3) $T \rightarrow T * F$
- (4) $T \rightarrow F$
- (5) $F \rightarrow (E)$
- (6) $F \rightarrow a$

- O estado inicial é o fechamento de $\{ S' \rightarrow \bullet E \}$.
- Como há um ponto antes de E, devemos acrescentar os itens $E \rightarrow \bullet E + T$ e $E \rightarrow \bullet T$.
- ... e por causa do ponto antes de T, acrescentamos $T \rightarrow \bullet T * F$ e $T \rightarrow \bullet F$.
- ... e por causa do ponto antes de F, acrescentamos $F \rightarrow \bullet (E)$ e $F \rightarrow \bullet a$.

Exemplo

$$(0) S' \rightarrow E$$

$$(1) E \rightarrow E + T$$

$$(2) E \rightarrow T$$

$$(3) T \rightarrow T * F$$

$$(4) T \rightarrow F$$

$$(5) F \rightarrow (E)$$

$$(6) F \rightarrow a$$

- Assim, o estado inicial (estado 0) é composto pelos itens:

$$S' \rightarrow \bullet E$$

$$E \rightarrow \bullet E + T$$

$$T \rightarrow \bullet T * F$$

$$F \rightarrow \bullet (E)$$

$$E \rightarrow \bullet T$$

$$T \rightarrow \bullet F$$

$$F \rightarrow \bullet a$$

À medida que a análise prossegue, o ponto deve caminhar para a direita nos diversos itens do estado.

Encontrado os estados

- Encontrado um símbolo X , passamos de um item $A \rightarrow \alpha \bullet X \beta$ para um item $A \rightarrow \alpha X \bullet \beta$.
 - Os símbolos terminais são encontrados na entrada; enquanto os símbolos não terminais são encontrados como produto de reduções.
- Se estamos em um estado p que tem itens com o ponto antes de um símbolo X , a transição com o símbolo X nos leva a outro estado q que tem um item com o ponto depois da ocorrência de X para cada item no estado p com o ponto antes de X . Os outros itens de q são itens obtidos por seu fechamento.
- A *tabela de transições* do analisador representa a *função de transição* δ , escrevemos $q = \delta(p)$.

Encontrado os estados

- Para gerar todos os estados do analisador, geramos todos os estados possíveis a partir do estado inicial e de outros estados gerados a partir dele.
 - Cada vez que um estado é obtido, verificamos se já ocorreu anteriormente.
- O número de estados (conjuntos de itens) é finito, uma vez que o número de itens é finito.

Exemplo

Estado 0:

$S' \rightarrow \bullet E$

$E \rightarrow \bullet E + T$

$E \rightarrow \bullet T$

$T \rightarrow \bullet T * F$

$T \rightarrow \bullet F$

$F \rightarrow \bullet (E)$

$F \rightarrow \bullet a$

Exemplo

Estado 0:

$S' \rightarrow \bullet E$

$E \rightarrow \bullet E + T$

$E \rightarrow \bullet T$

$T \rightarrow \bullet T * F$

$T \rightarrow \bullet F$

$F \rightarrow \bullet (E)$

$F \rightarrow \bullet a$

E



Estado 1:

$S' \rightarrow E \bullet$

$E \rightarrow E \bullet + T$

Exemplo

Estado 0:

$S' \rightarrow \bullet E$

$E \rightarrow \bullet E + T$

$E \rightarrow \bullet T$

$T \rightarrow \bullet T * F$

$T \rightarrow \bullet F$

$F \rightarrow \bullet (E)$

$F \rightarrow \bullet a$

E



Estado 1:

$S' \rightarrow E \bullet$

$E \rightarrow E \bullet + T$

Estado 2:

$E \rightarrow T \bullet$

$T \rightarrow T \bullet * F$

T



Exemplo

Estado 0:

$S' \rightarrow \bullet E$

$E \rightarrow \bullet E + T$

$E \rightarrow \bullet T$

$T \rightarrow \bullet T * F$

$T \rightarrow \bullet F$

$F \rightarrow \bullet (E)$

$F \rightarrow \bullet a$

E



Estado 1:

$S' \rightarrow E \bullet$

$E \rightarrow E \bullet + T$

T



Estado 2:

$E \rightarrow T \bullet$

$T \rightarrow T \bullet * F$

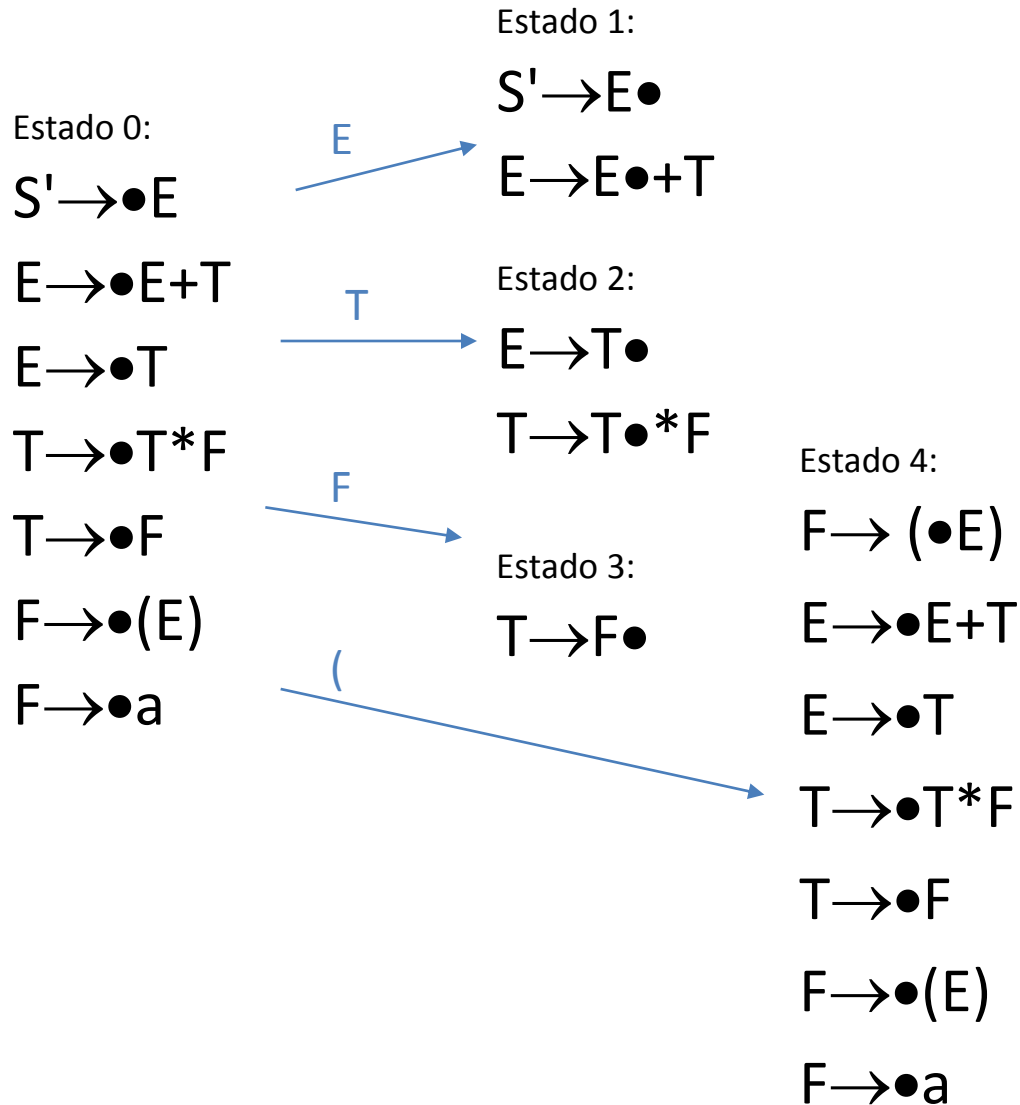
F



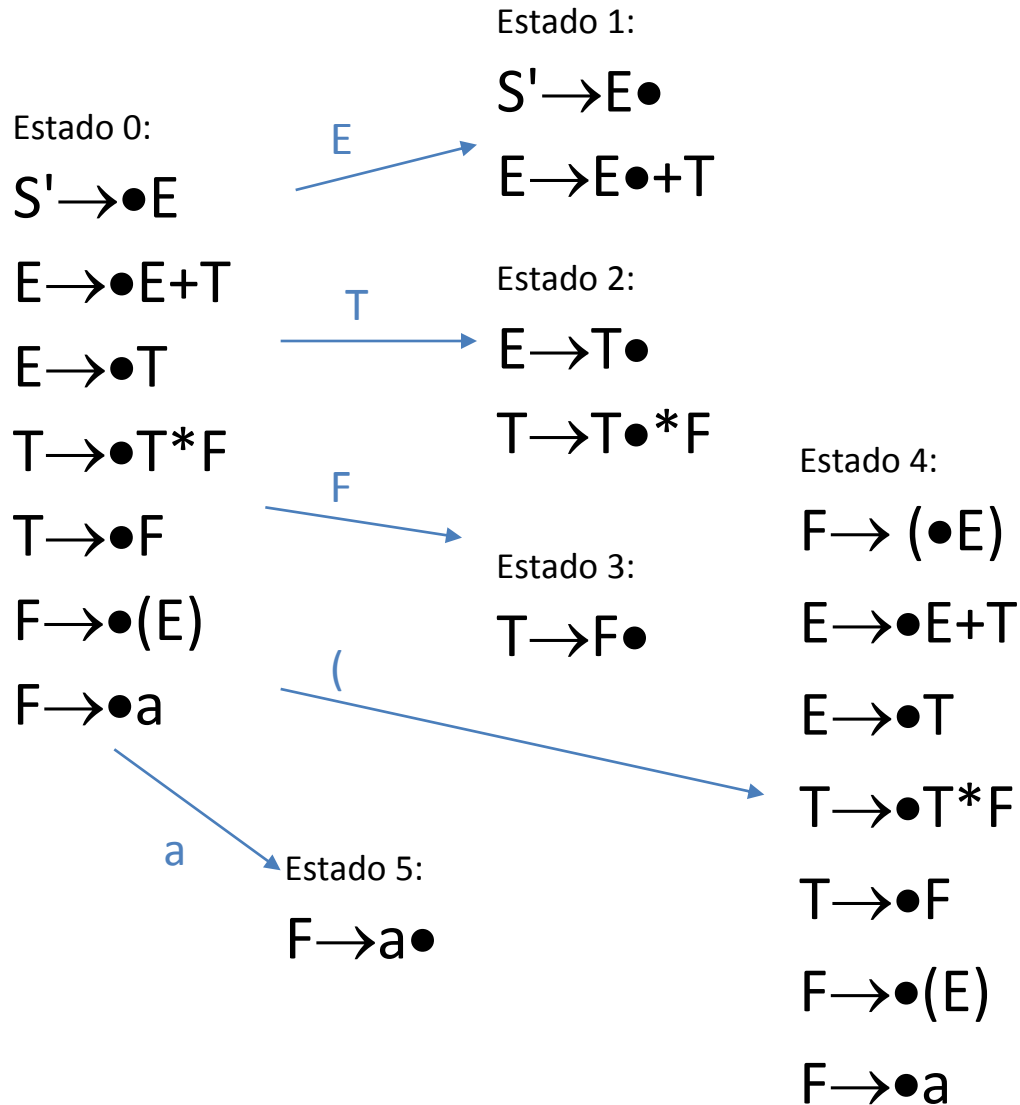
Estado 3:

$T \rightarrow F \bullet$

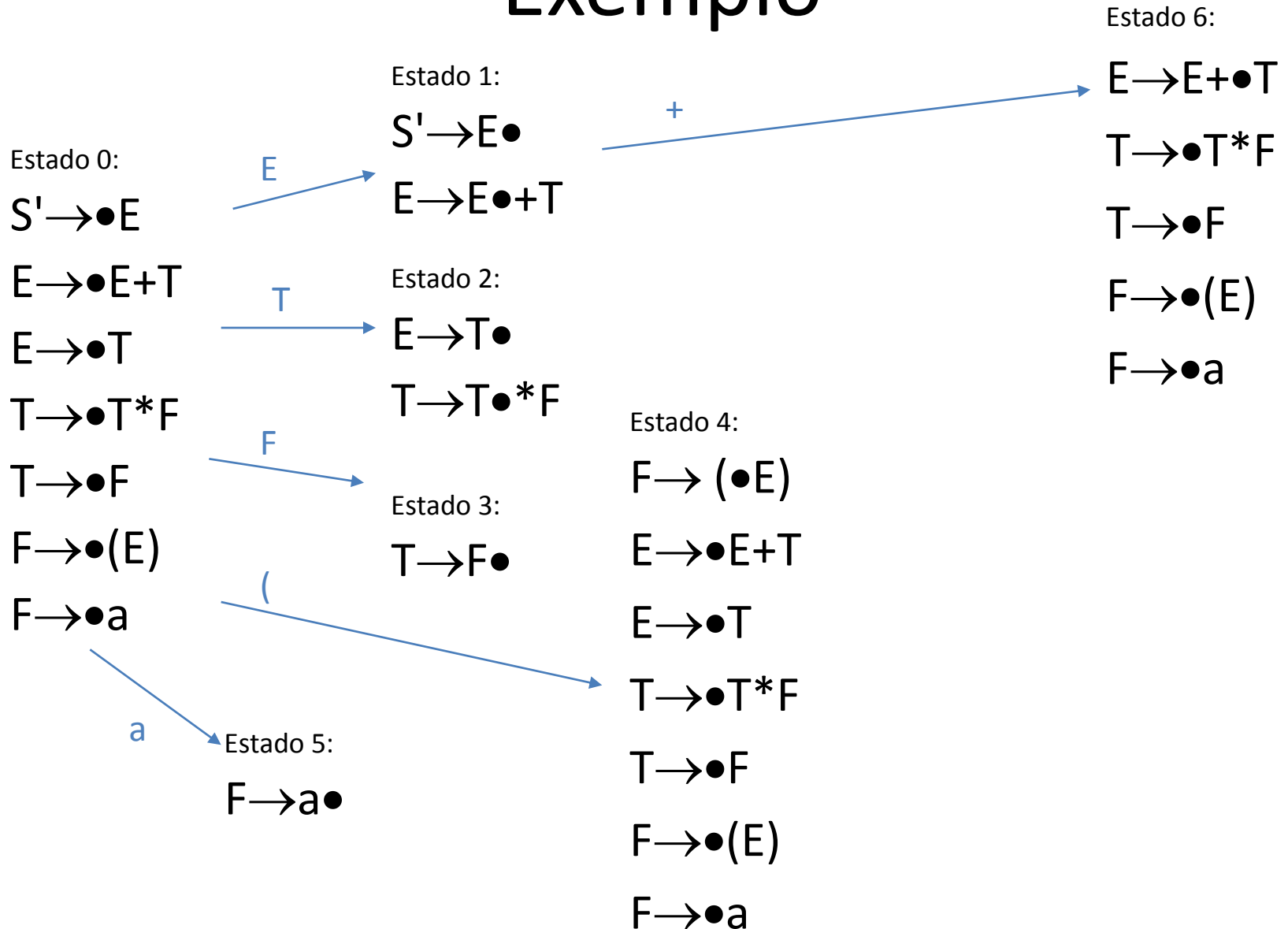
Exemplo



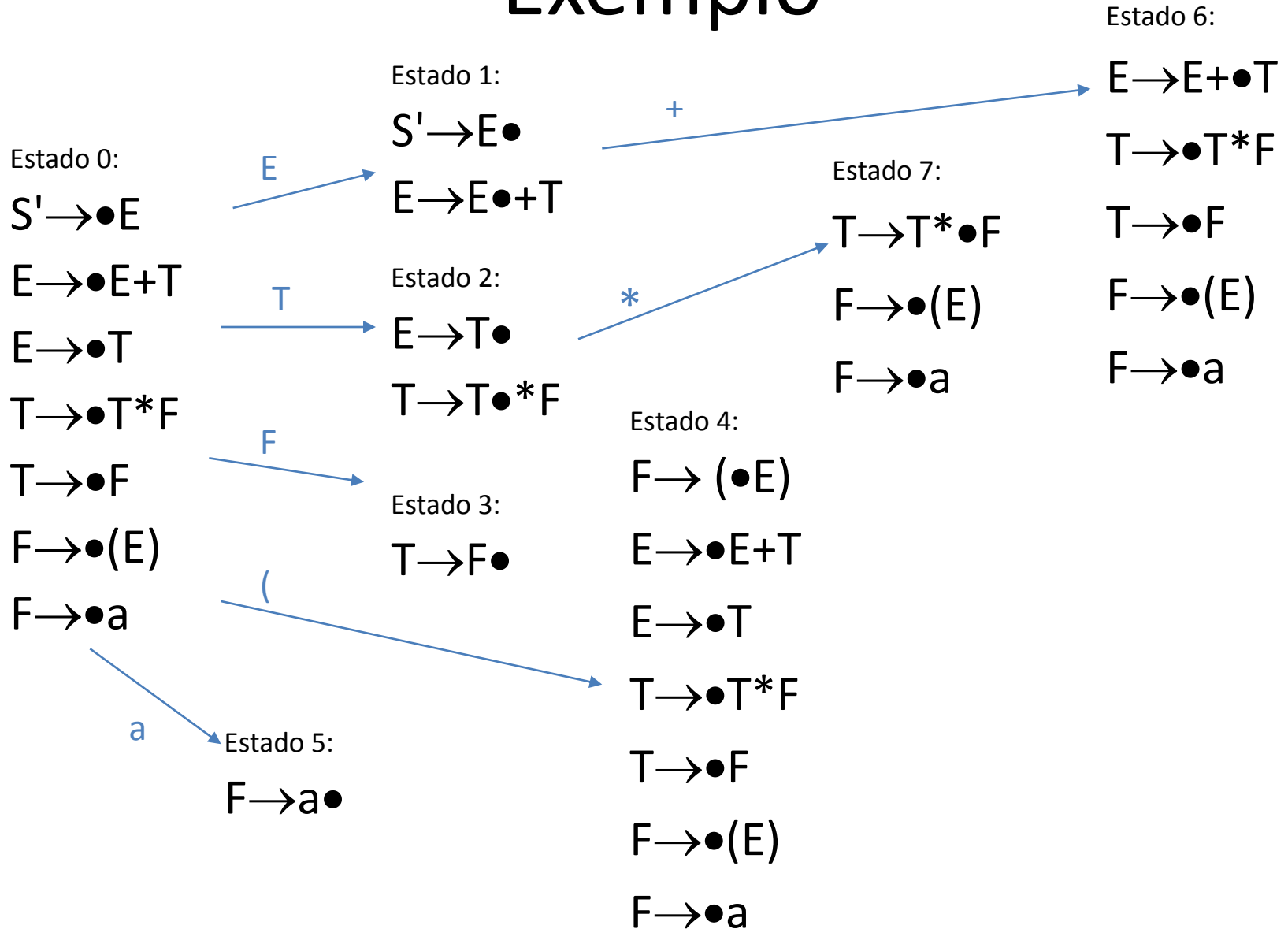
Exemplo



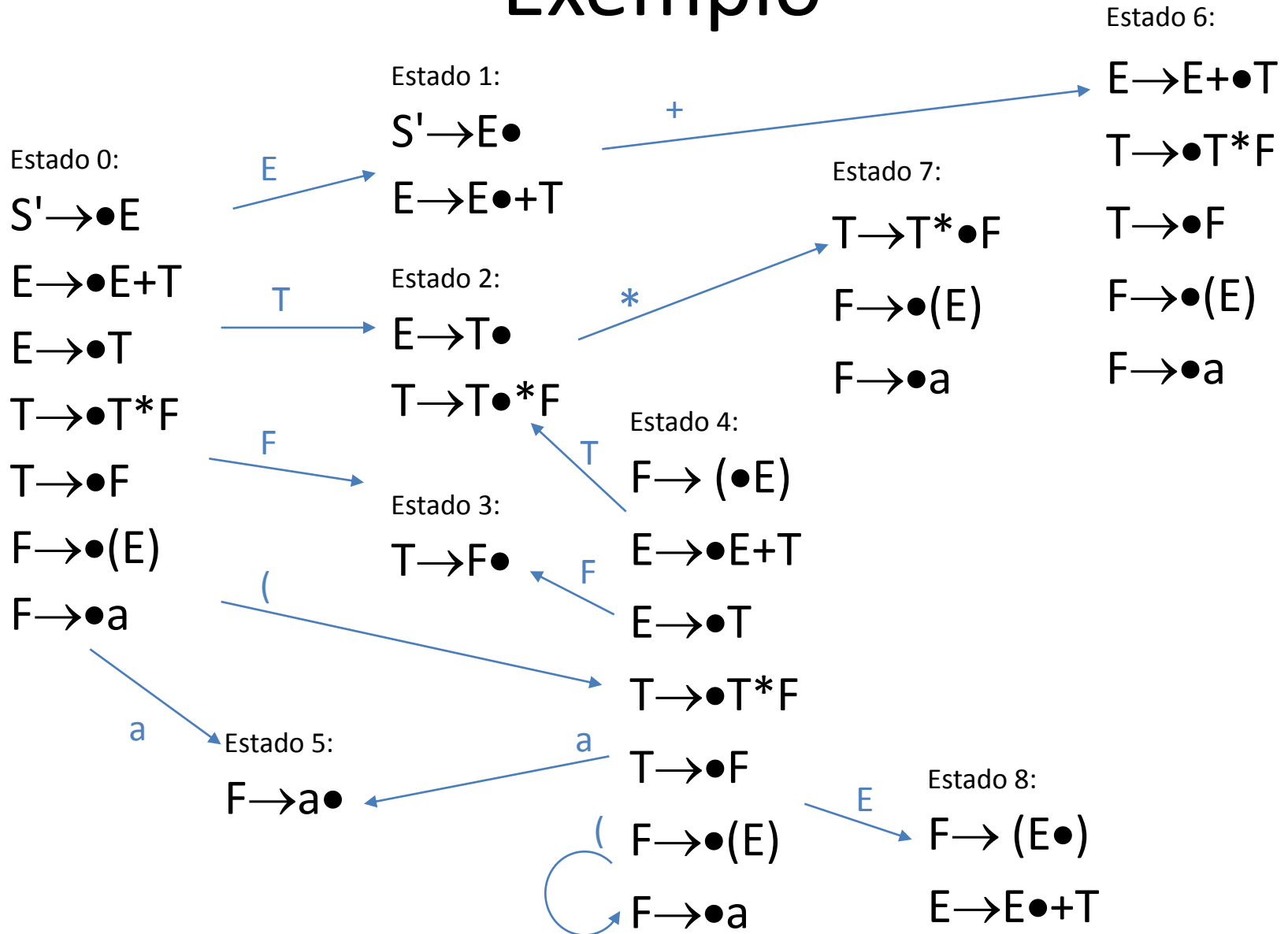
Exemplo



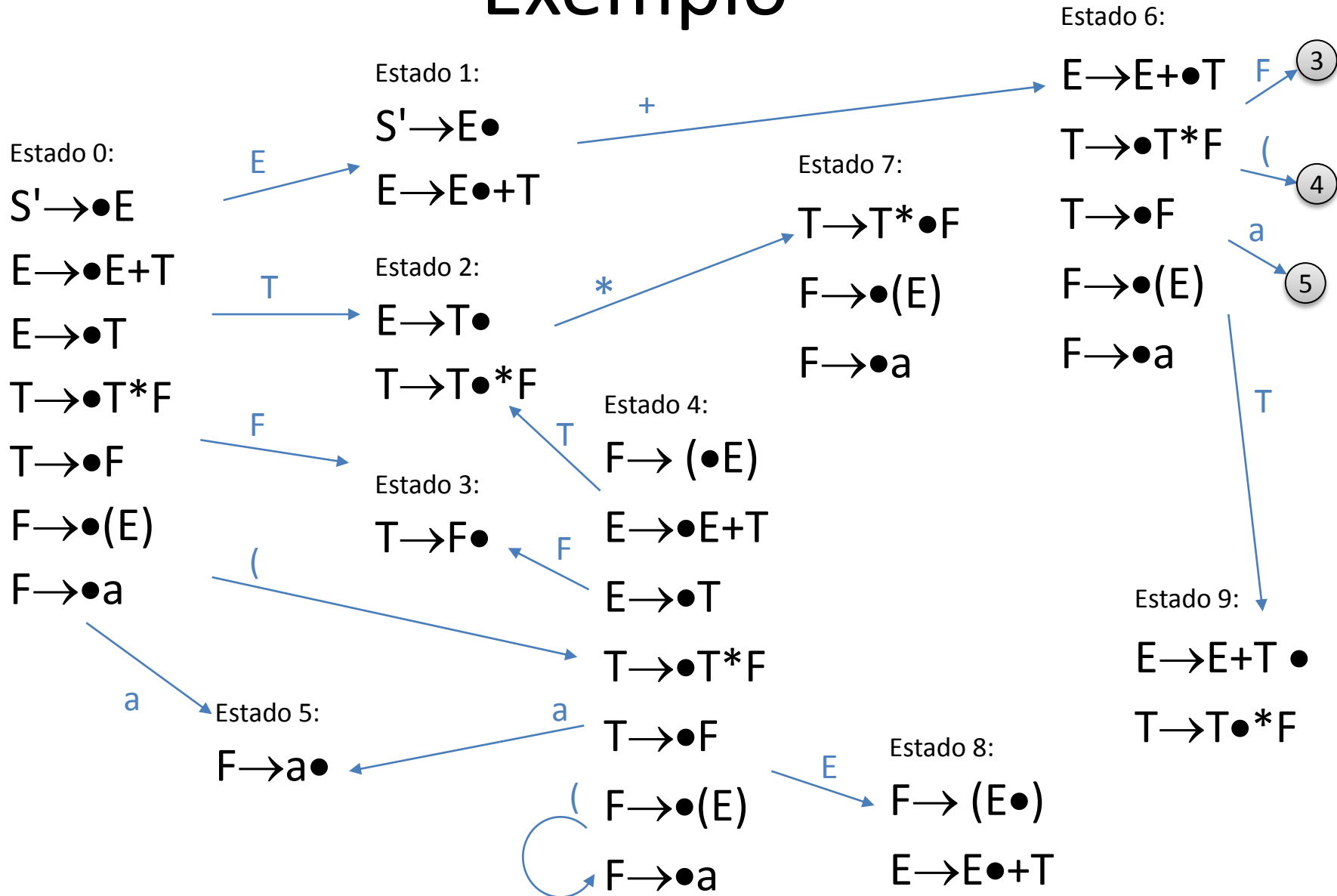
Exemplo



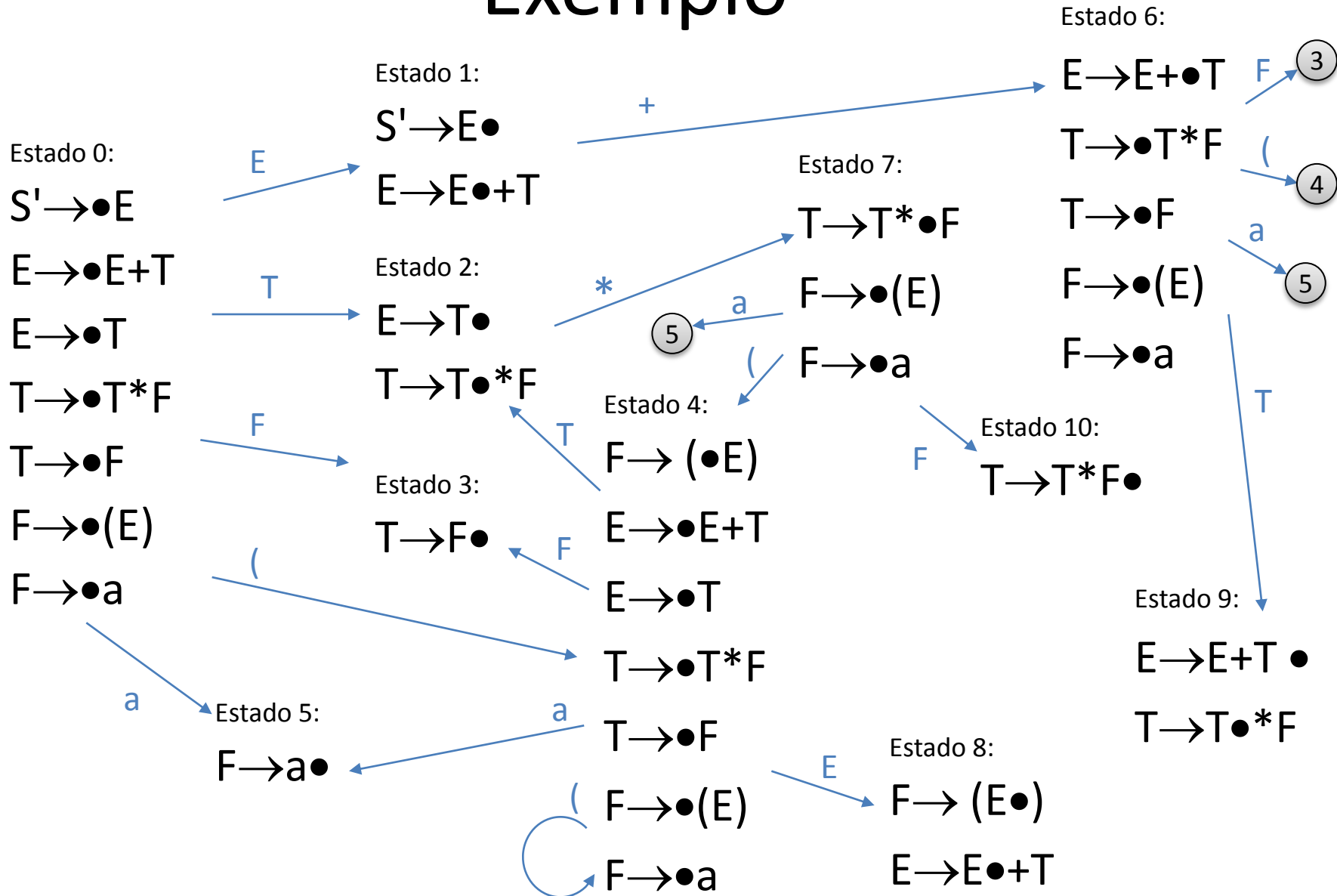
Exemplo



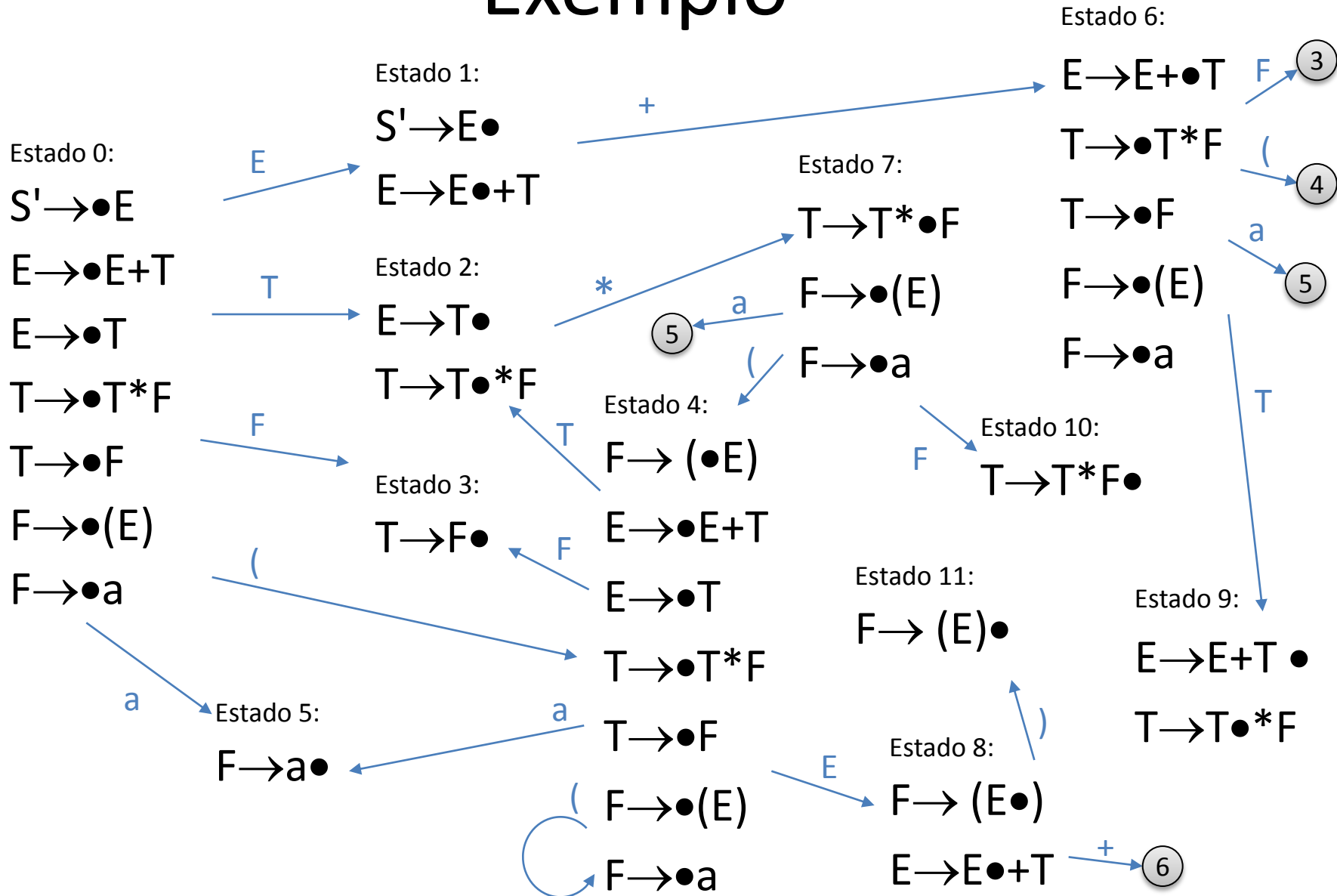
Exemplo



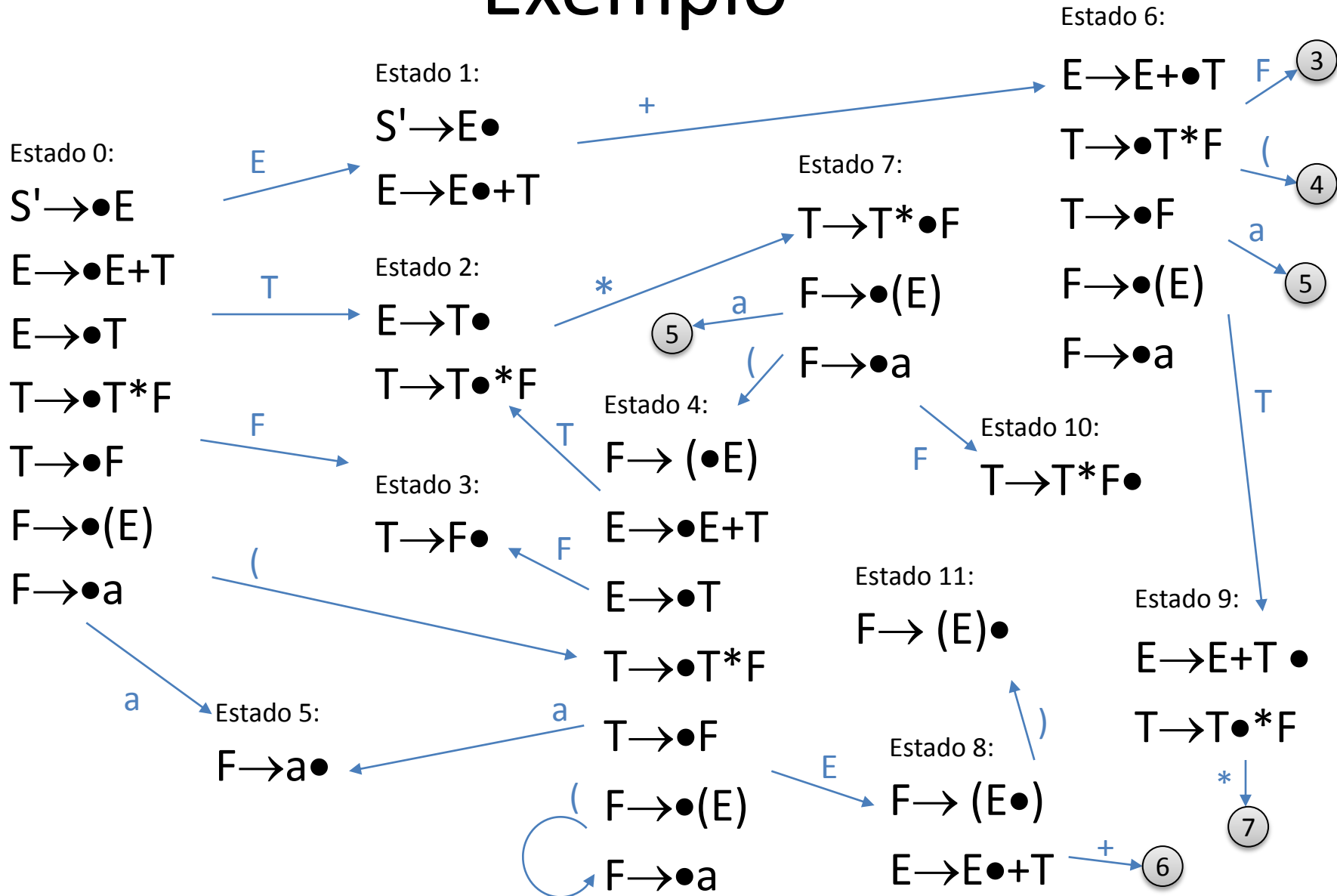
Exemplo



Exemplo



Exemplo



Note que ...

- O analisador sLR(1) é um analisador ascendente. Em vez de símbolos, a pilha do analisador contém os estados correspondentes aos símbolos.
- Primeiro, observamos que a cada estado q , com exceção do estado inicial, corresponde exatamente um símbolo X , que é o único símbolo que ocorre depois do ponto, nos itens do estado q .
- Todas as transições para q são feitas com o símbolo X , podendo ocorrer, entretanto, que dois ou mais estados sejam acessíveis pelo mesmo símbolo X . Neste caso, os estados se distinguem por conter informação adicional sobre a posição em que o símbolo X ocorre na cadeia de entrada.

Ações do analisador sLR(1)

- Um **empilhamento** (*shift*) pode ocorrer quando existe uma *transição com um terminal* a partir do estado corrente (o estado do topo da pilha).
- Quando existe um item completo $B \rightarrow \gamma \bullet$, no estado corrente, pode ser feita uma redução pela regra $B \rightarrow \gamma$. Em um analisador sLR(1), a regra é:
“reduza pela regra $B \rightarrow \gamma$ se o símbolo da entrada pertencer ao $\text{Follow}(B)$.”
- Esse primeiro símbolo (do resto) da entrada é conhecido como o símbolo de *lookahead*.

Ações do analisador sLR(1)

Assim, a tabela do analisador sLR(1) pode conter as seguintes ações (em função do estado q do topo da pilha e do símbolo s de lookahead):

- **Empilhamento**, o empilhamento do estado p (que representa s) deve ser empilhado, e o analisador léxico deve ser acionado para obter outro símbolo da entrada.
- **Redução**, se $T[q, s] = \textit{reduce } B \rightarrow \gamma$, os $|\gamma|$ estados correspondentes a γ devem ser retirados da pilha, e o estado $\delta(q, B)$ deve ser empilhado, representando B .
- **Aceitação**, se $T[q, s] = \textit{reduce } S' \rightarrow S$, o processo se encerra com sucesso.

Tabela do analisador sLR(1)

	E	T	F	(a	+	*)	\$
0	1	2	3	4	5	-	-	-	-
1	-	-	-	-	-	6	-	-	<i>R0</i>
2	-	-	-	-	-	<i>R2</i>	7	<i>R2</i>	<i>R2</i>
3	-	-	-	-	-	<i>R4</i>	<i>R4</i>	<i>R4</i>	<i>R4</i>
4	8	2	3	4	5	-	-	-	-
5	-	-	-	-	-	<i>R6</i>	<i>R6</i>	<i>R6</i>	<i>R6</i>
6	-	9	3	4	5	-	-	-	-
7	-	-	10	4	5	-	-	-	-
8	-	-	-	-	-	6	-	11	-
9	-	-	-	-	-	<i>R1</i>	7	<i>R1</i>	<i>R1</i>
10	-	-	-	-	-	<i>R3</i>	<i>R3</i>	<i>R3</i>	<i>R3</i>
11	-	-	-	-	-	<i>R5</i>	<i>R5</i>	<i>R5</i>	<i>R5</i>

Exemplo

- Notação utilizada no exemplo:
 - *shift* q (ação de empilhamento): número do estado q
 - *reduce* $B \rightarrow \gamma$ (ação de redução), representada por ri , onde i é o número da regra;
 - ação de parada será indicada como $r0$.
- Cadeia: $(a+a) * a$

Exemplo: $(a+a)*a$

Pilha	(resto da) entrada	Ação
0	$(a+a)*a$	Empilhar: 4
0 (4	$a+a)*a$	Empilhar: 5
0 (4 a 5	$+a)*a$	Reduzir: 6
0 (4 F 3	$+a)*a$	Reduzir: 4
0 (4 T 2	$+a)*a$	Reduzir: 2
0 (4 E 8	$+a)*a$	Empilhar: 6
0 (4 E 8 + 6	$a)*a$	Empilhar: 5
0 (4 E 8 + 6 a 5	$)*a$	Reduzir: 6
0 (4 E 8 + 6 F 3	$)*a$	Reduzir: 4
0 (4 E 8 + 6 T 9	$)*a$	Reduzir: 1
0 (4 E 8	$)*a$	Empilhar: 11
0 (4 E 8) 11	$*a$	Reduzir: 5
0 F 3	$*a$	Reduzir: 4
0 T 2	$*a$	Empilhar: 7

Exemplo: $(a+a) * a$

[illegible]