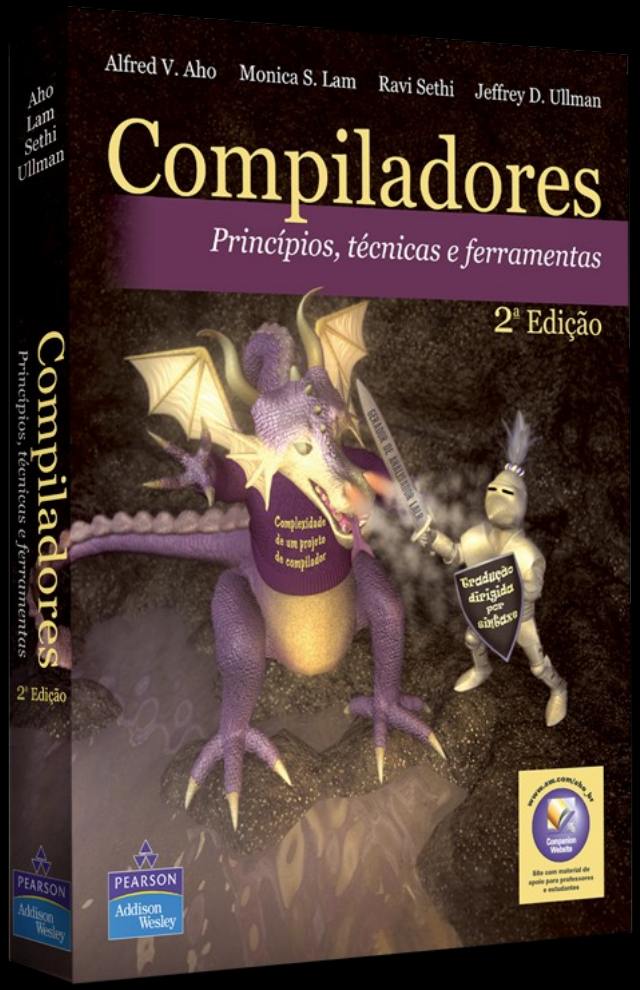
Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman

Compiladores

Princípios,técnicas e ferramentas

2ª Edição



Capítulo 4

Análise Sintática

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

Análise Sintática

- Estuda métodos usados para decidir se uma cadeia pertence ou não à linguagem definida por uma gramática.

- o objetivo é: sendo a cadeia pertencente à linguagem, obter a estrutura sintática da mesma, em forma de árvore de derivação

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Análise Sintática

- observação: a análise sintática pode não construir explicitamente a árvore de derivação, mais as ações conceitualmente realizam essa tarefa.

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

Análise Sintática

- Métodos mais usados

- ascendentes (“bottom-up”) - começam pelas folhas

- descendentes (“top-down”) - começam pela raiz

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Um exemplo de uma gramática

expression	→	*expression* + *term*
expression	→	*expression* − *term*
expression	→	*term*
term	→	*term* * factor
term	→	*term* / factor
term	→	*factor*
factor	→	(*expression*)
factor	→	id

FIGURA 4.2 Gramática para expressões aritméticas simples.

Onde: t = id + - * / ( )

Nt = expression term factor

S = expression

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Derivações

Figura 4.3 - derivação para a gramática

E → E + E | E * E | -E | (E) | **id**

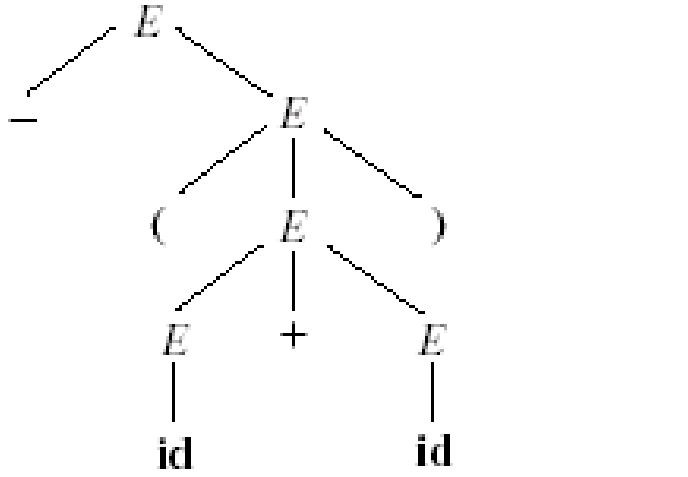


FIGURA 4.3 Árvore de derivação para −(id + id).

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Derivações

Figura 4.4 – a sequência da derivação anterior

E → E + E | E * E | -E | (E) | **id**

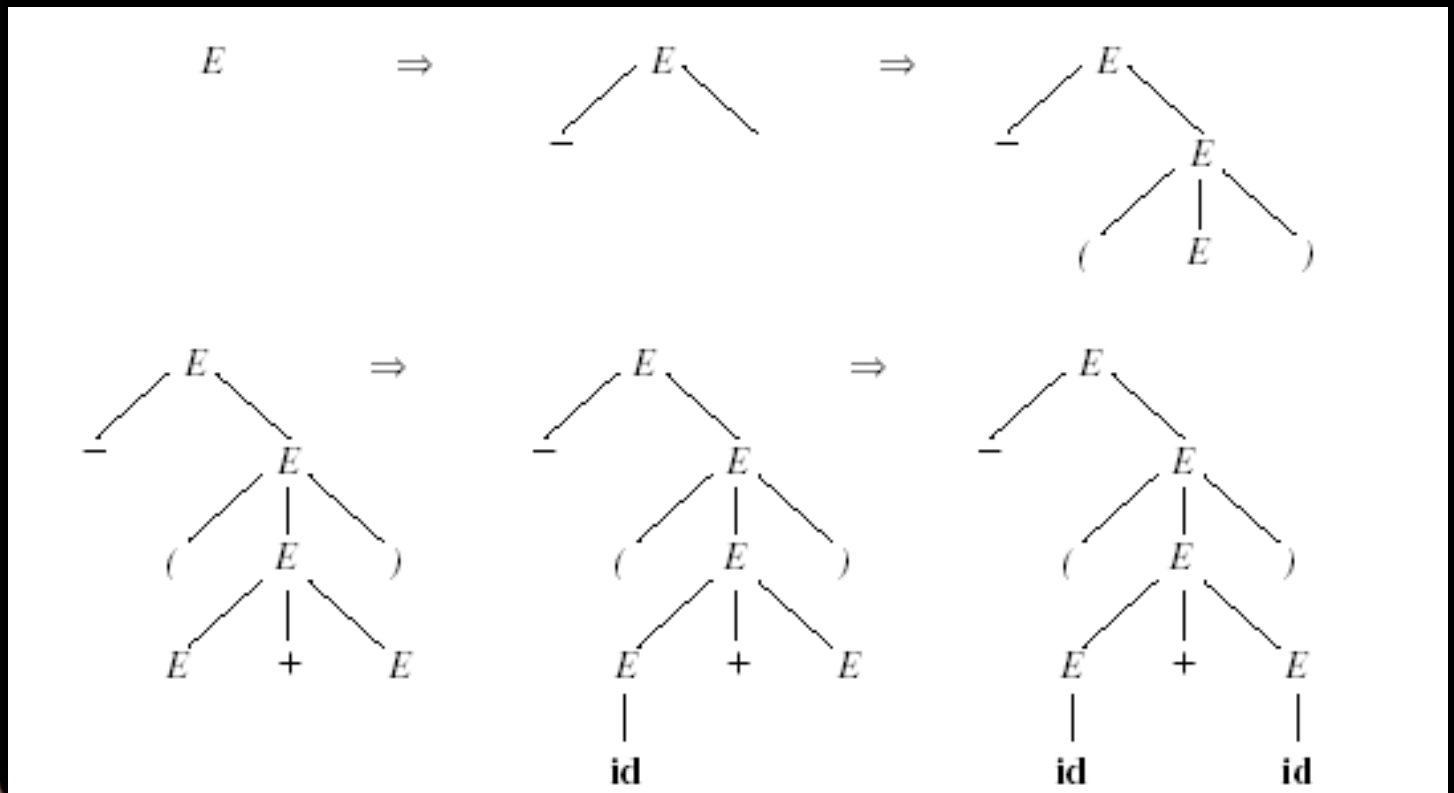


FIGURA 4.4 Seqüência de árvores de derivação para a derivação (4.8).

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

Figura 4.5 – um exemplo de ambiguidade

E → E + E | E * E | -E | (E) | **id**

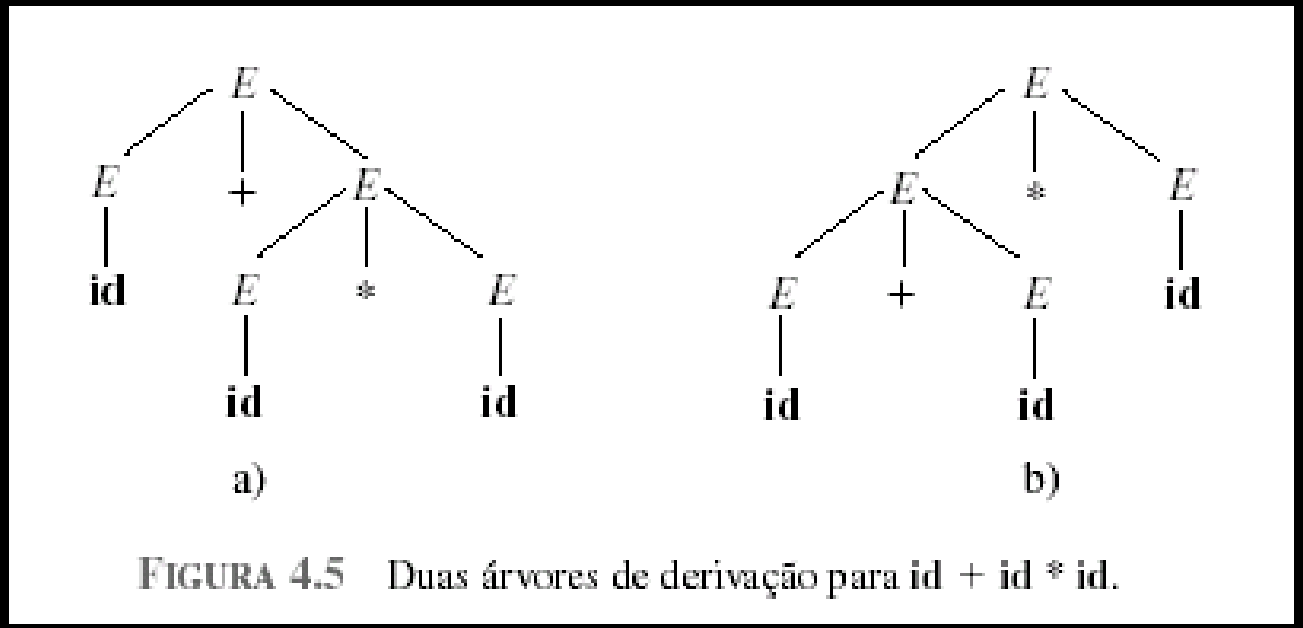


FIGURA 4.5 Duas árvores de derivação para id + id * id.

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Análise sintática descendente – um exemplo para a gramática

E → TE'
E' → +TE' | ε
T → FT'
T' → *FT' | ε
F → (E) | **id**

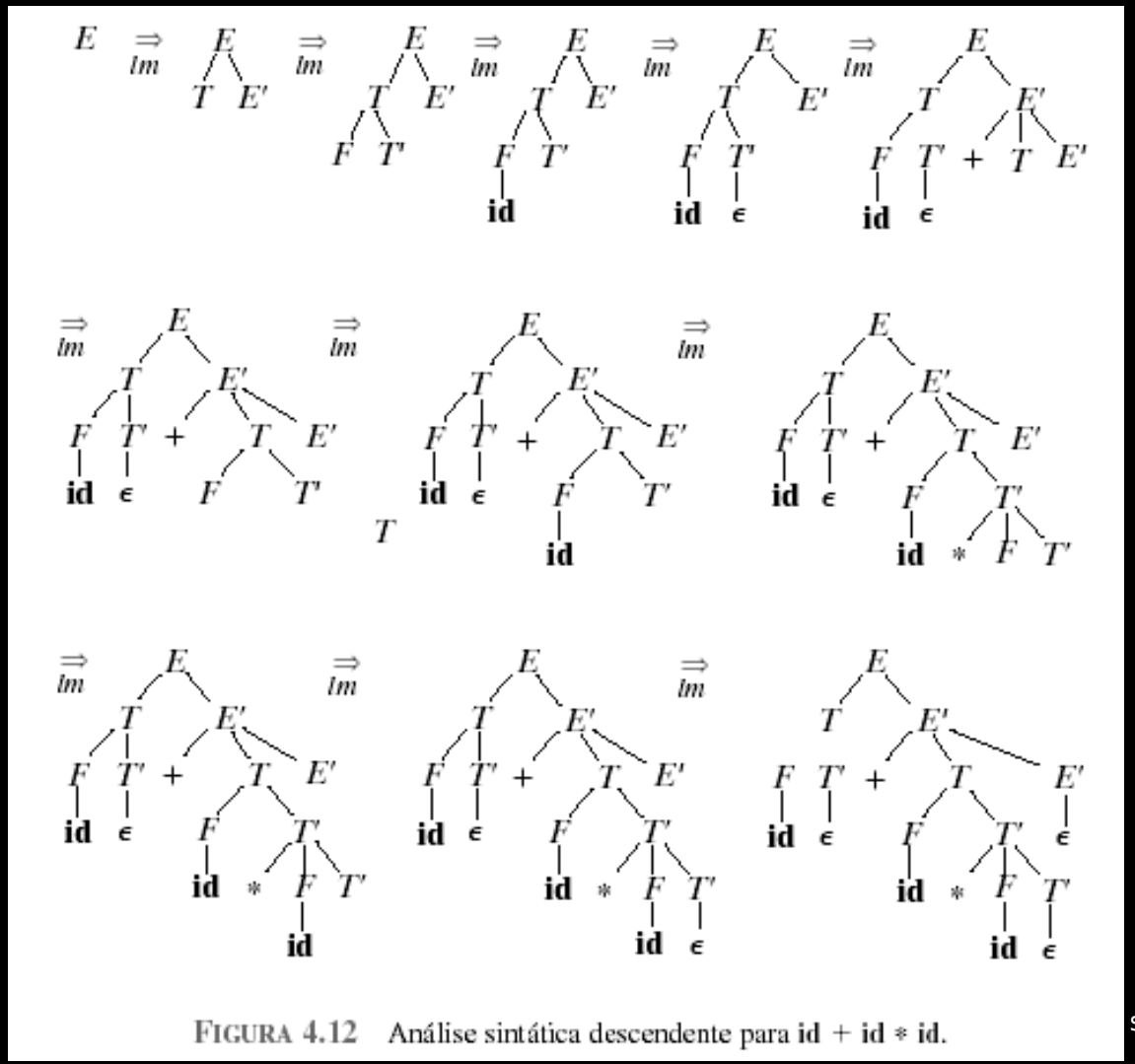


FIGURA 4.12 Análise sintática descendente para id + id * id.

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

```
        void A() {
1)          Escolha uma produção-A, A → X1 X2 ··· Xk;
2)          for ( i = 1 até k ) {
3)                  if ( Xi é um não-terminal )
4)                          ativa procedimento Xi();
5)                  else if ( Xi igual ao símbolo de entrada a )
6)                          avance na entrada para o próximo símbolo terminal;
7)                  else /* ocorreu um erro */;
            }
        }
```

FIGURA 4.13 Um procedimento típico para um não-terminal em um analisador descendente.

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Analise Sintática Descendente

1. Análise Descendente com Retrocesso

- também chamado “tentativa e erro”
- um dos primeiros métodos de Análise Sintática
- ineficiência em memória e tempo
- parecido com o processo de derivação de árvores sintáticas.

Analise Sintática Descendente com retrocesso

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Método:

1. dada a cadeia inicial α, e folha S (símbolo inicial).

2. seja X rótulo da folha corrente

- se X for não-terminal, escolhe-se uma produção $X \rightarrow X1$ X2..Xn e substitui na árvore de derivação D. X1 passa a ser a folha corrente e o passo 2 repete.

se X for terminal, e $\alpha = X\beta$ para alguma cadeia β, então adota-se β como novo valor de α. Pega-se então, a próxima folha em D (da esquerda para a direita) para ser a folha corrente, e repete passo2.

Analise Sintática Descendente com retrocesso

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

(2.cont.):

... se X é terminal e o primeiro símbolo de α não é X (ou $\alpha = \lambda$), então deve-se retroceder à última configuração em que foi feita a escolha de uma produção, adotando-se uma outra alternativa para o não-terminal da folha corrente...

Analise Sintática Descendente com retrocesso

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

(2.cont.):

... caso não haja mais alternativas, repete-se o retrocesso até que seja encontrada, e volta a repetir o passo 2. Se o algoritmo avança além da última folha de D, mas $\alpha \neq \lambda$, então deve-se retroceder como no caso anterior.

Analise Sintática Descendente com retrocesso

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

3. A análise termina quando o algoritmo avança além da última folha de D e $\alpha = \lambda$. Nesse caso, a cadeia dada é uma sentença da linguagem. Se o algoritmo é forçado a retroceder à configuração inicial do passo1 depois de esgotar todas as alternativas para o símbolo inicial da gramática, a cadeia não pertence à linguagem.

Analise Sintática Descendente com retrocesso

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Ex: Considere a gramática

$E \rightarrow T + E \mid T$
$T \rightarrow F * T \mid F$
$F \rightarrow a \mid b \mid (E)$

e a sentença **a*b**

temos a seguinte sequência de derivação, onde ↑ indica folha corrente e x⇐, significa retroceder à derivação de número x.

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

a*b

E

↑

(1)

a*b

E

T + E

↑

(2)

a*b

E

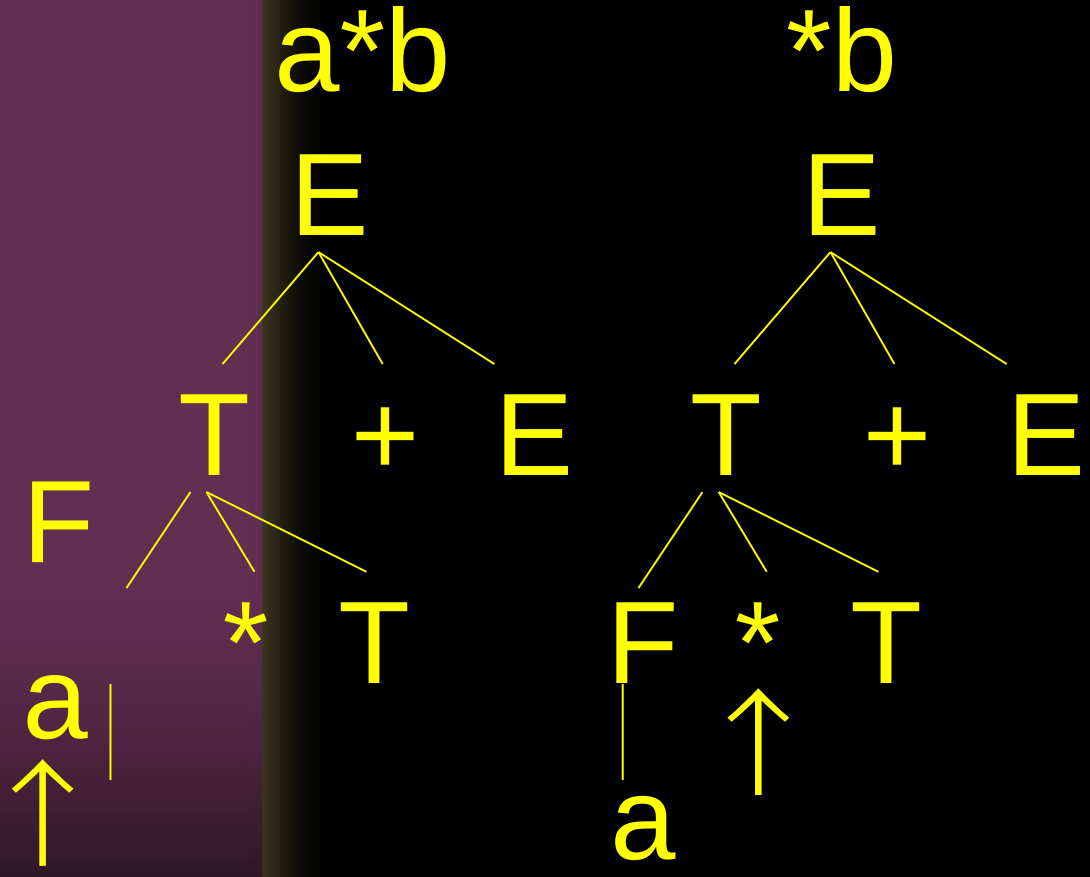
T + E

F * T

↑

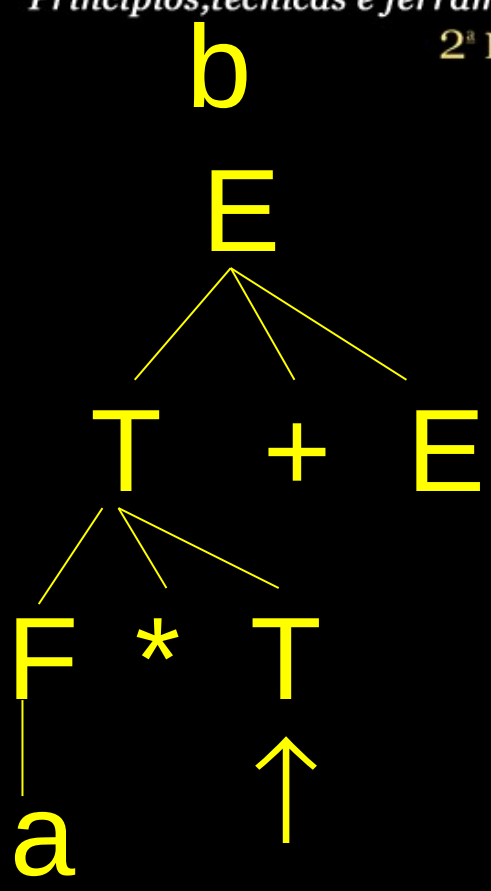
(3)

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

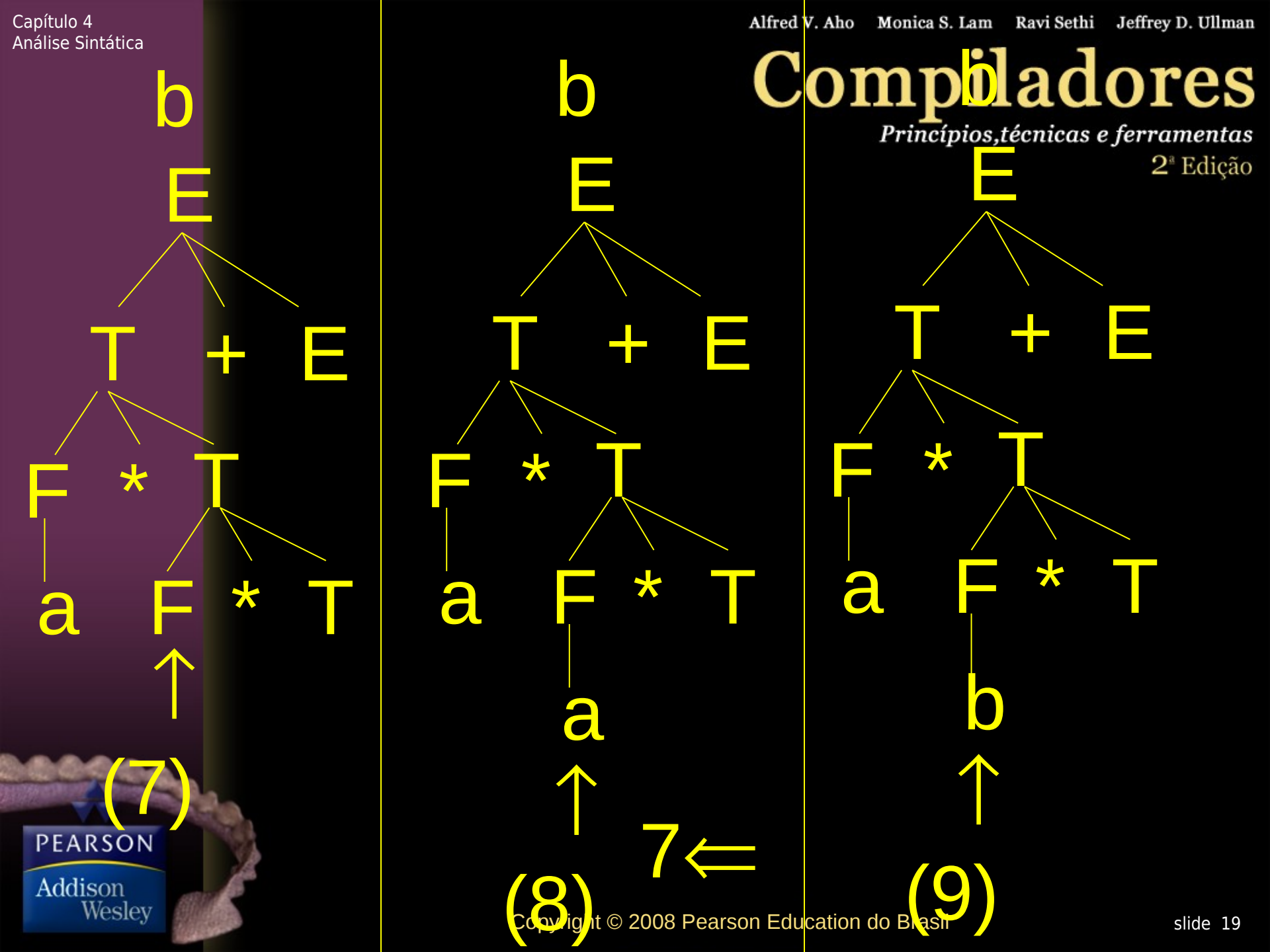


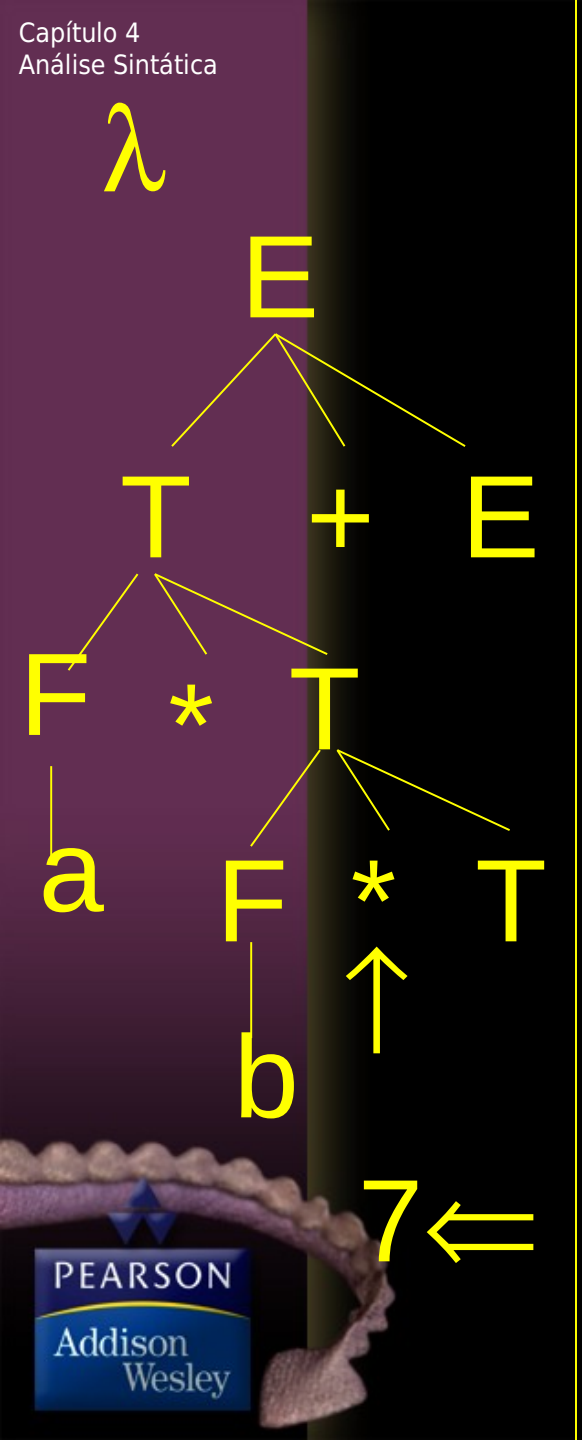
(4)

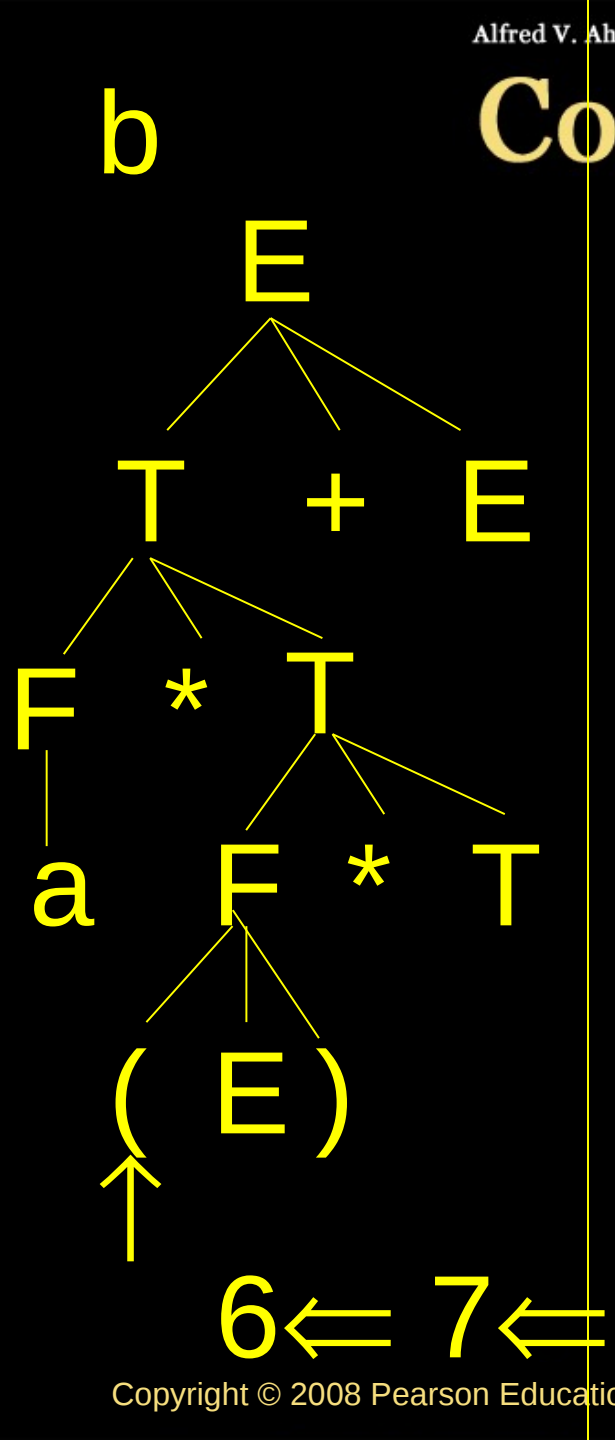
(5)

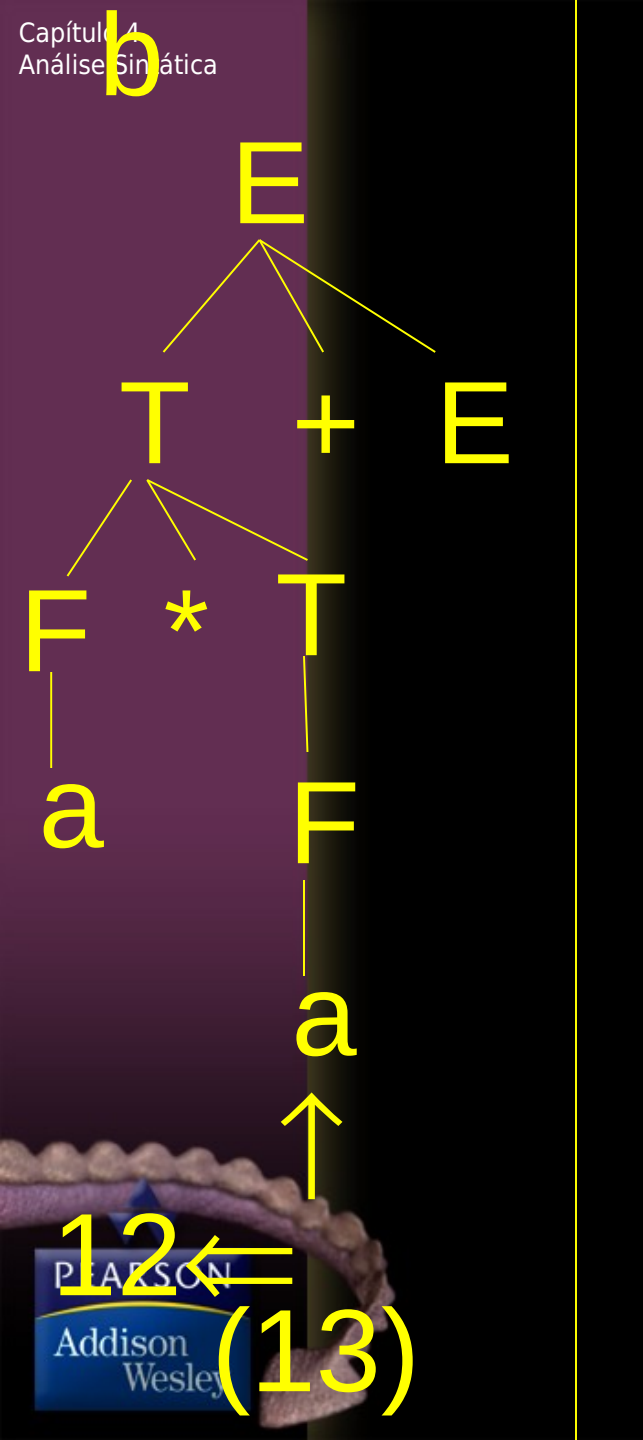


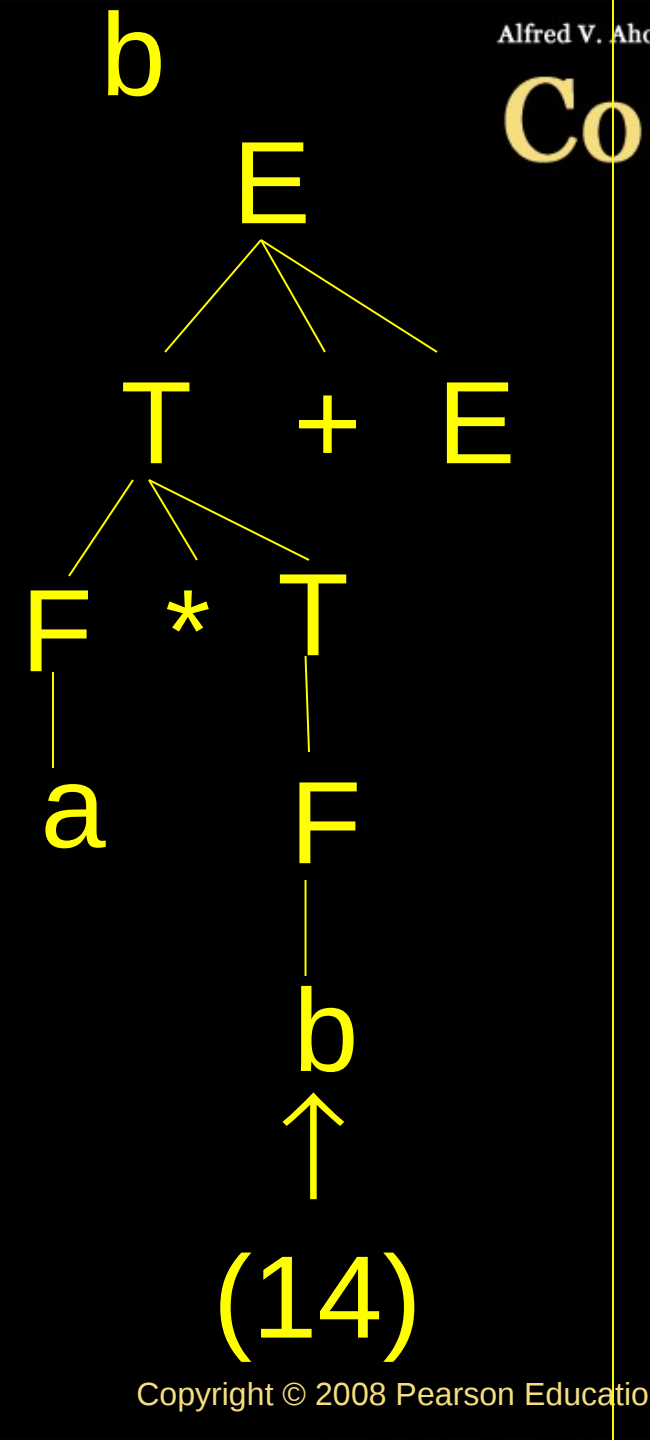
(6)

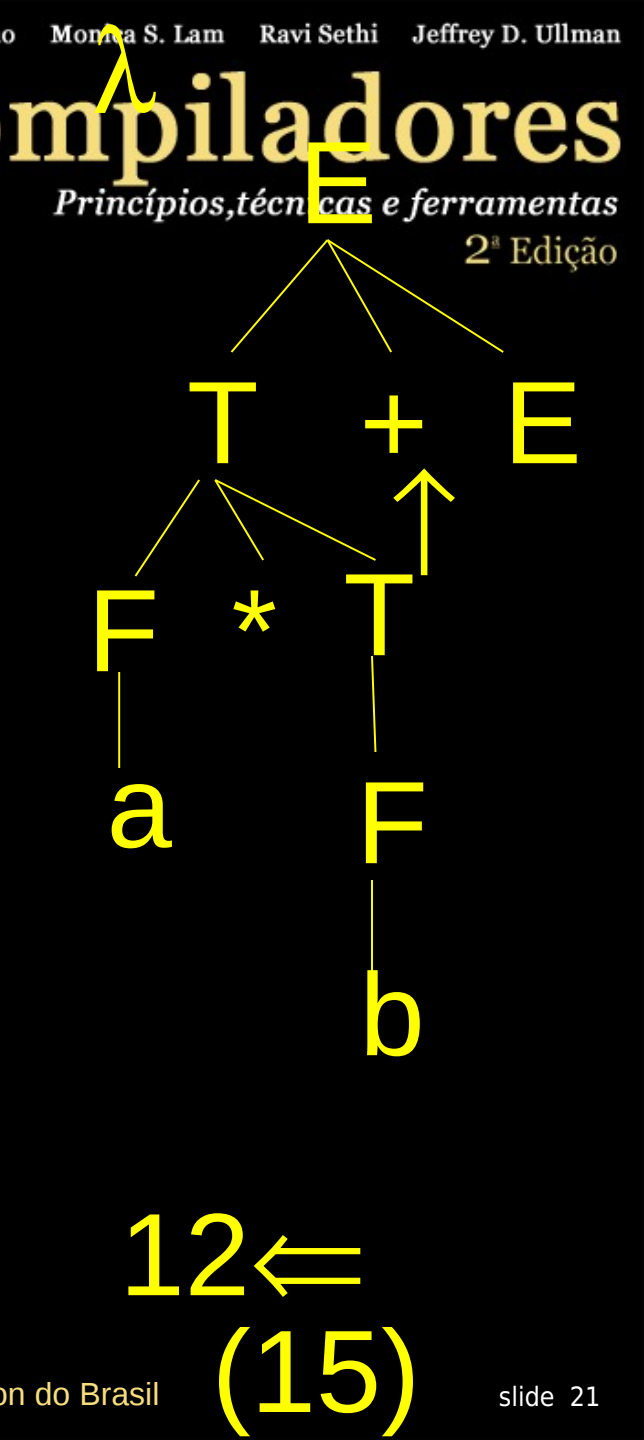






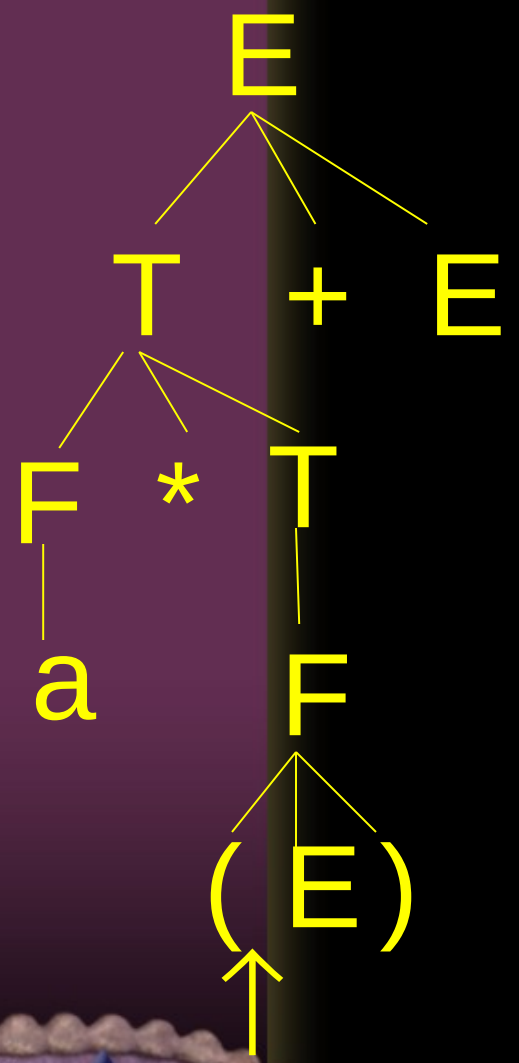






Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

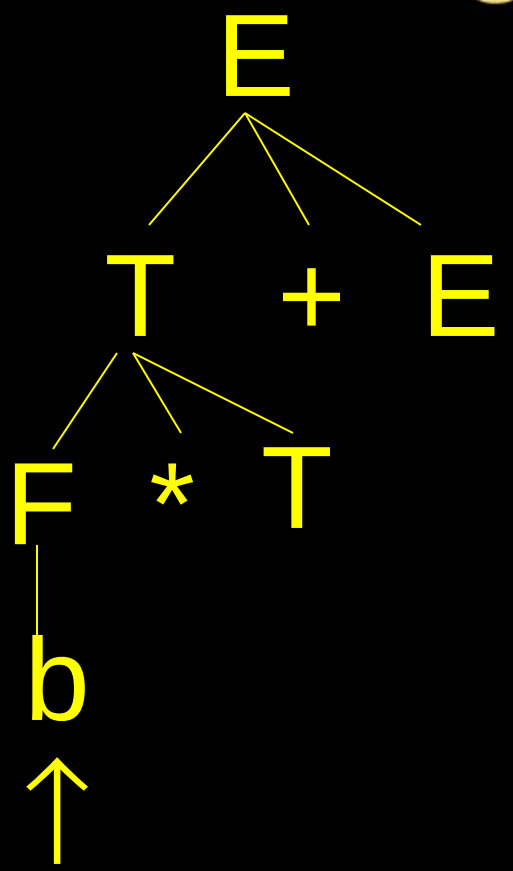
λ



3⇐6⇐12

(16)

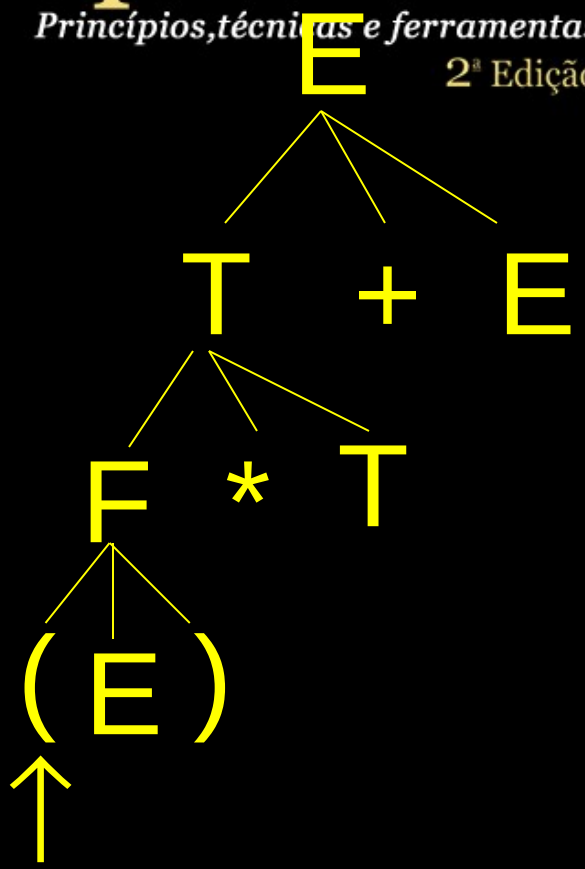
a*b



3⇐

(17)

a*b

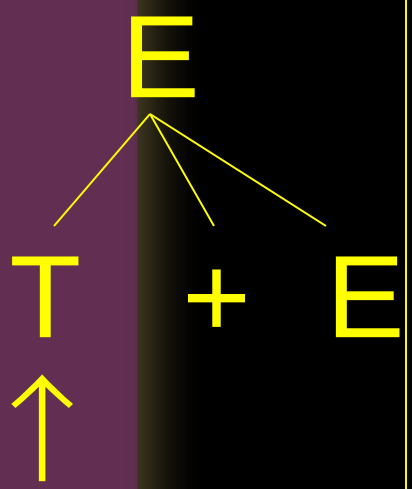


2⇐ 3⇐

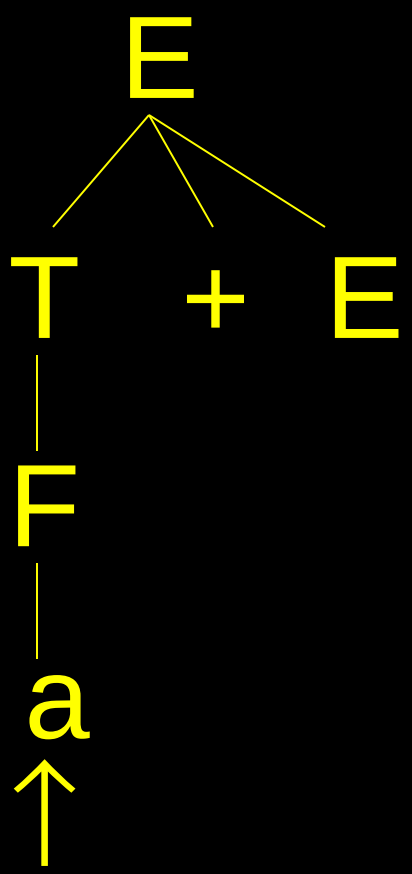
(18)

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

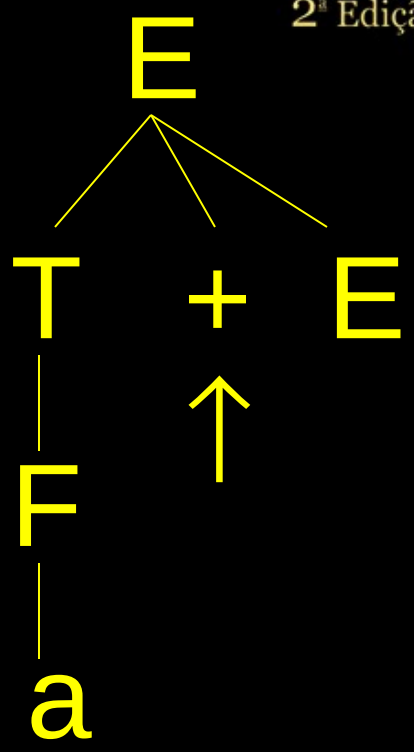
a*b



a*b



*b

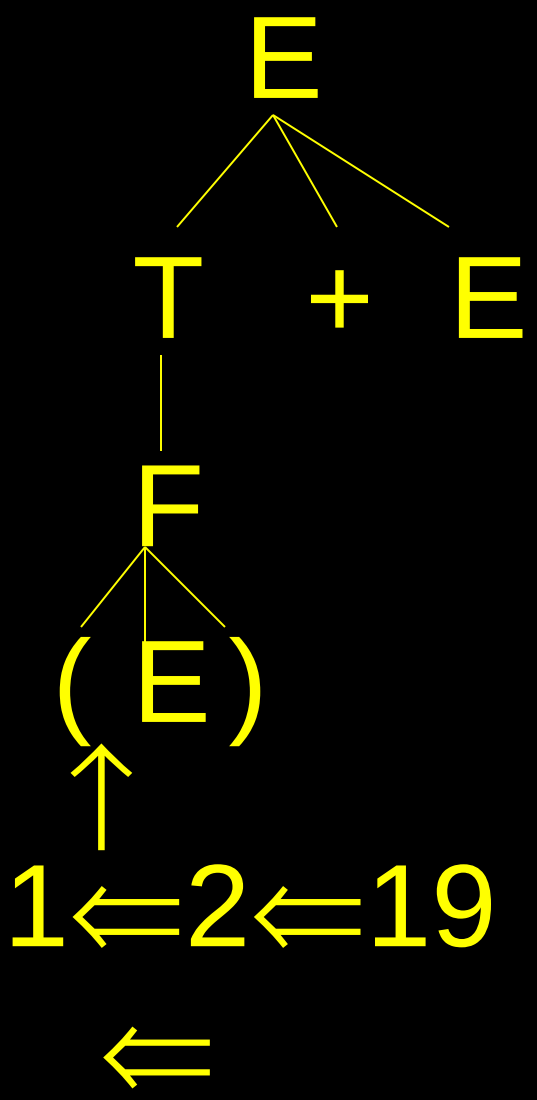


19⇐

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

a*b

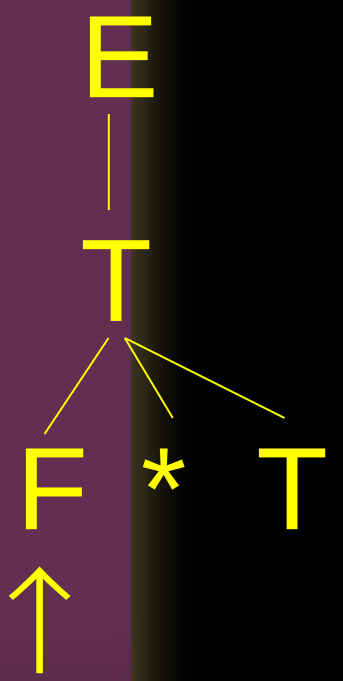
a*b



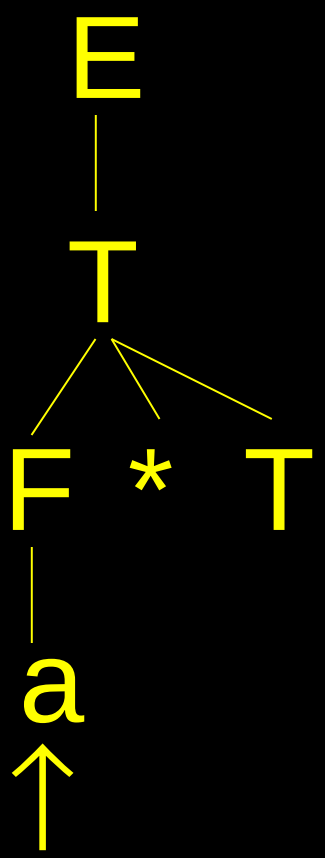
a*b

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

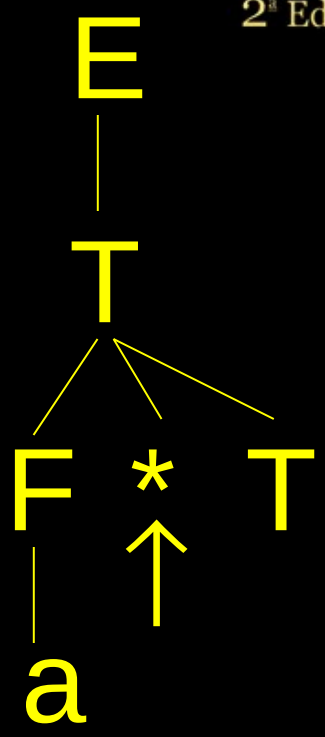
a*b

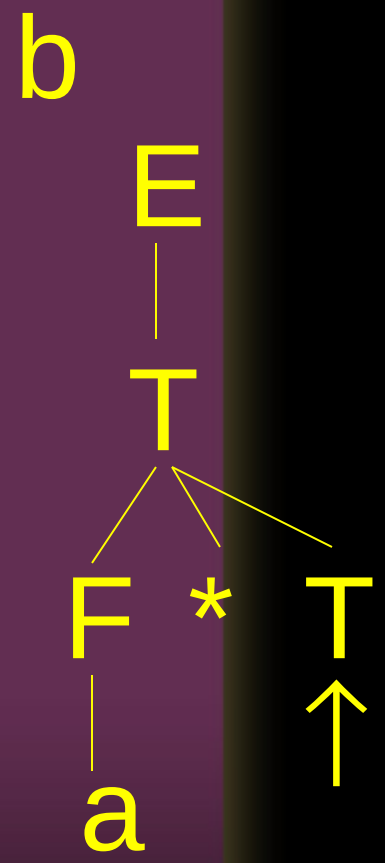


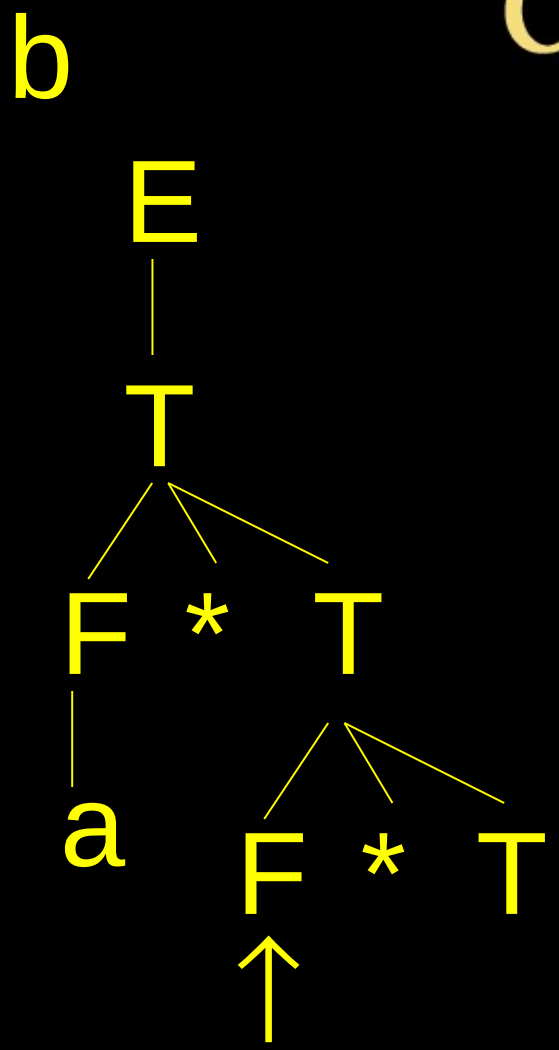
a*b

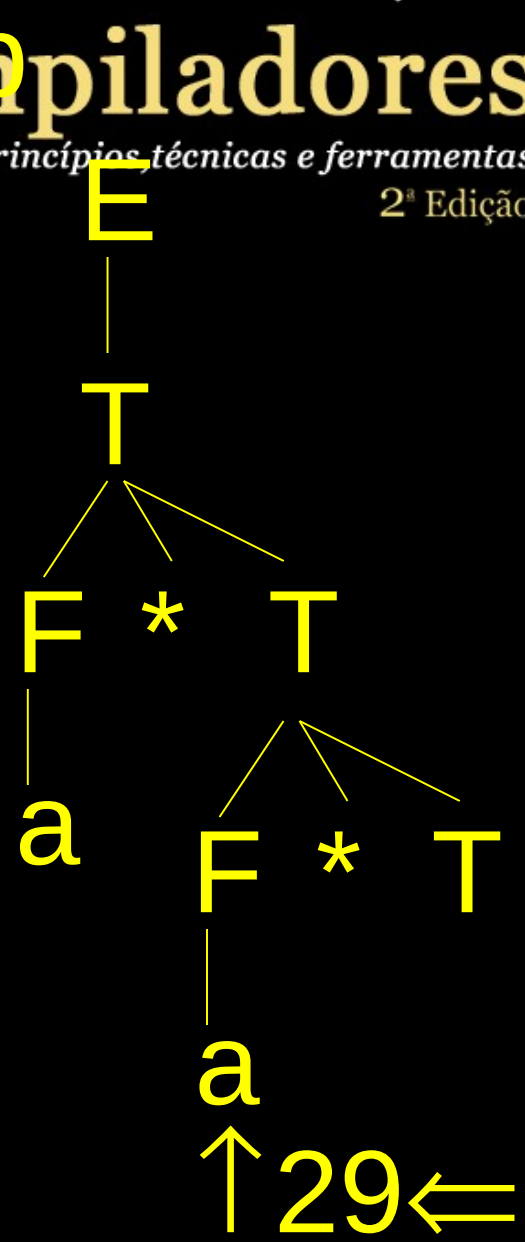


*b



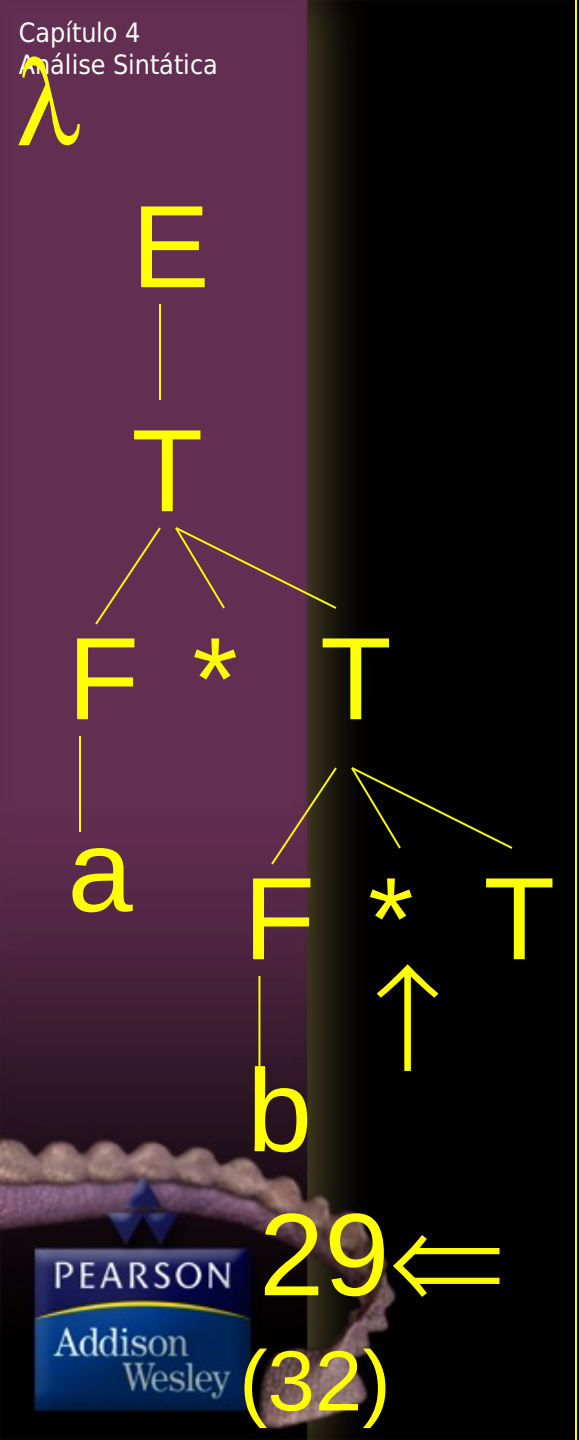


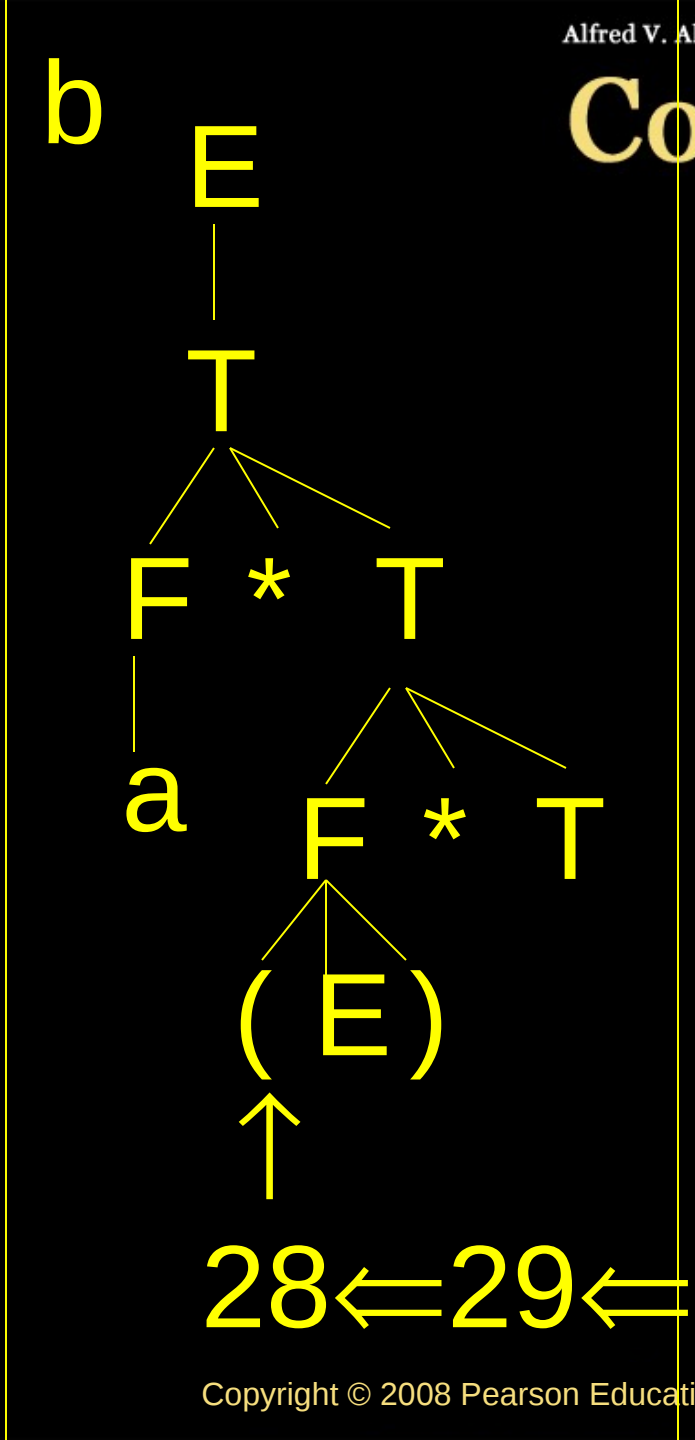


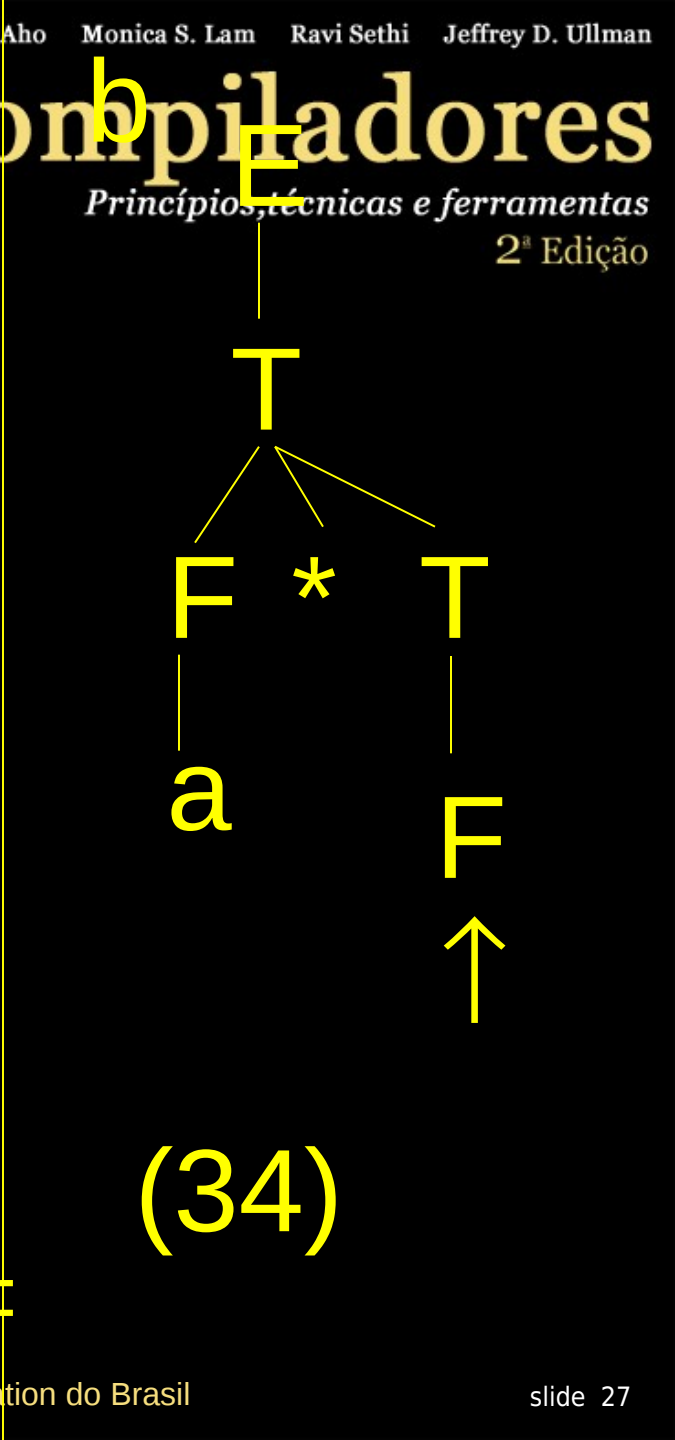


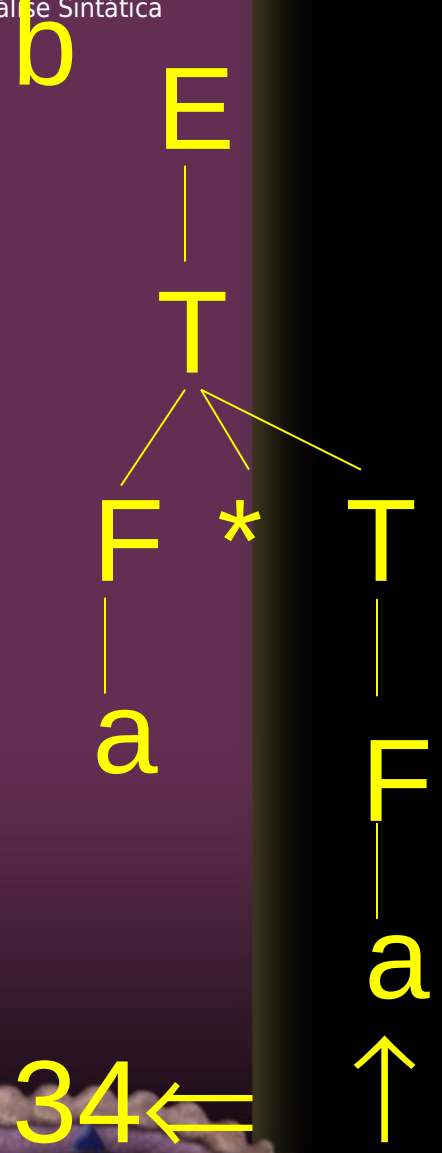
Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição



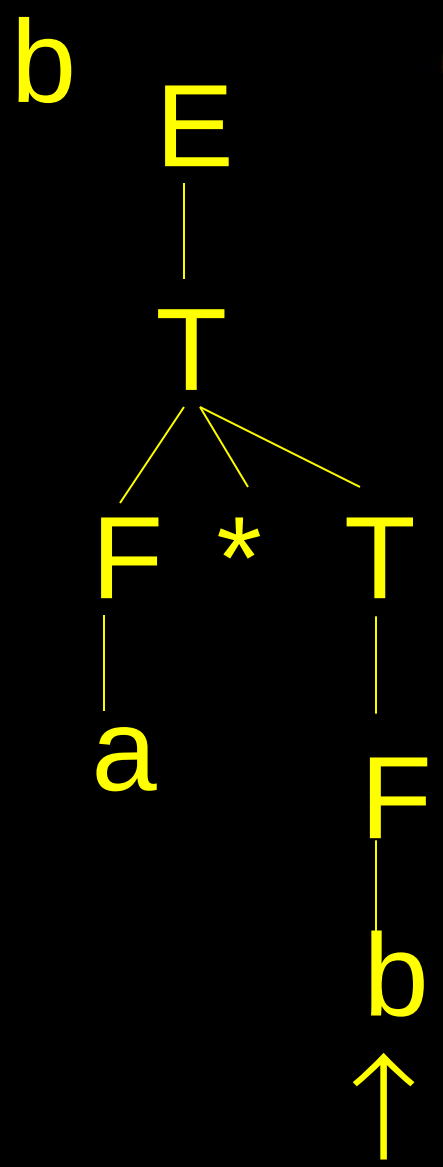




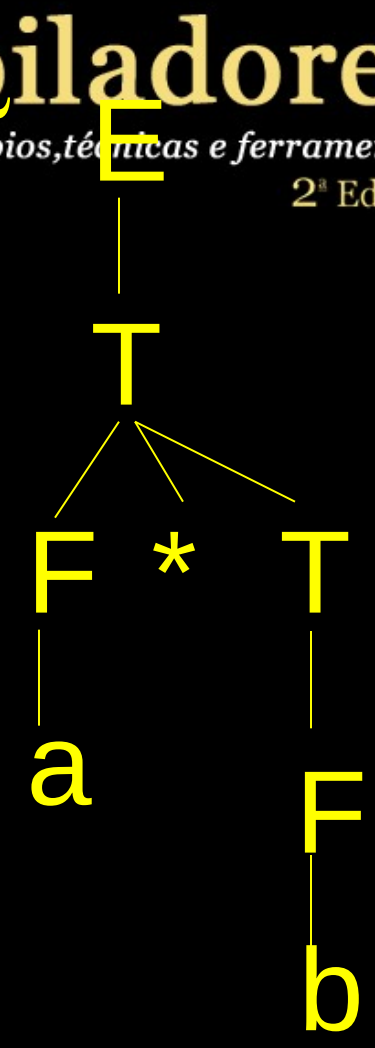


34⇐

(35)



(36)



(37)

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Exemplo2: dada a gramática

G = {Vn, Vt, S, P}

onde Vn={S,A,B}

Vt = {a,b,c,d}

P: $S \rightarrow A$

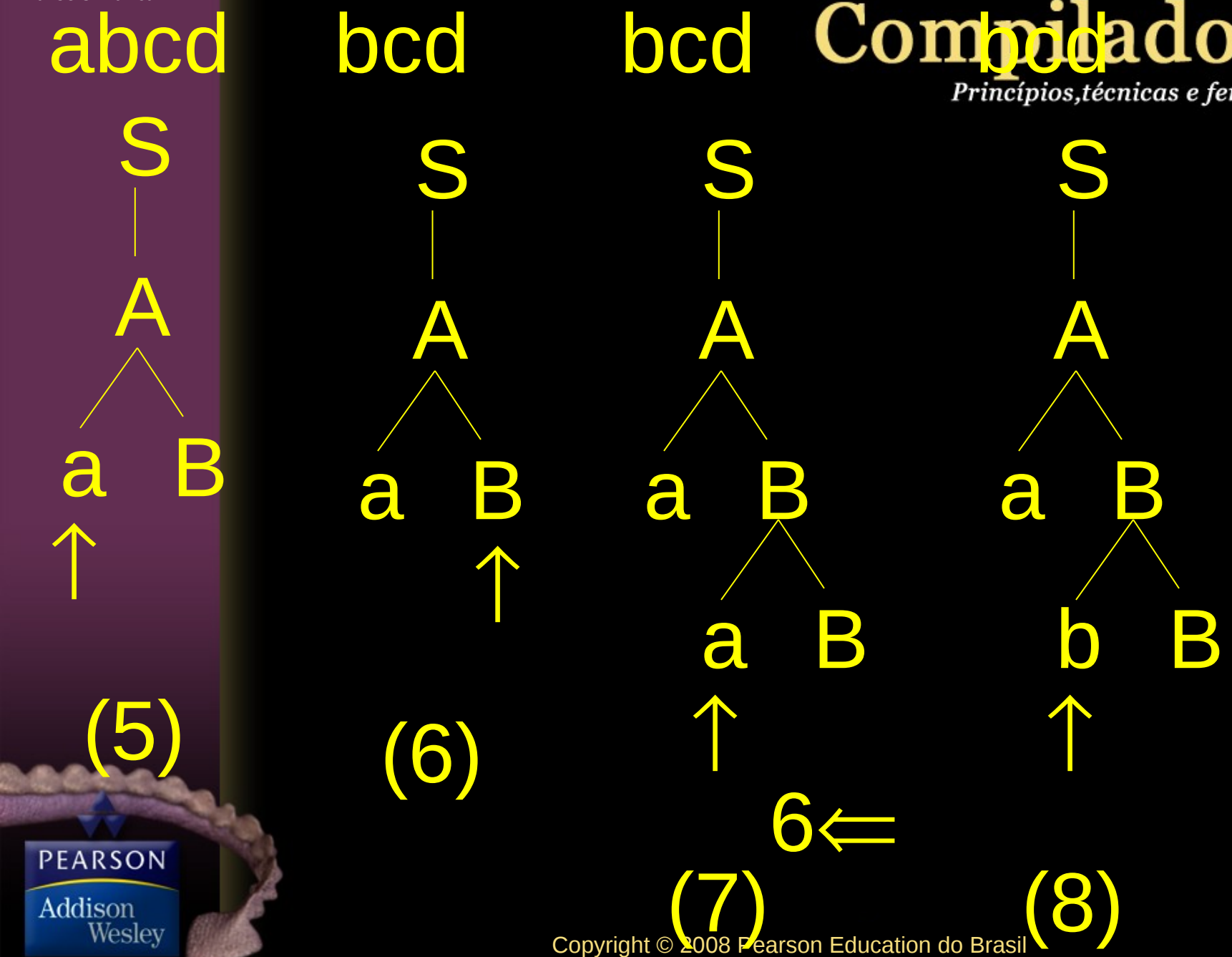
$A \rightarrow a \mid aB$

$B \rightarrow bB \mid cB \mid d$

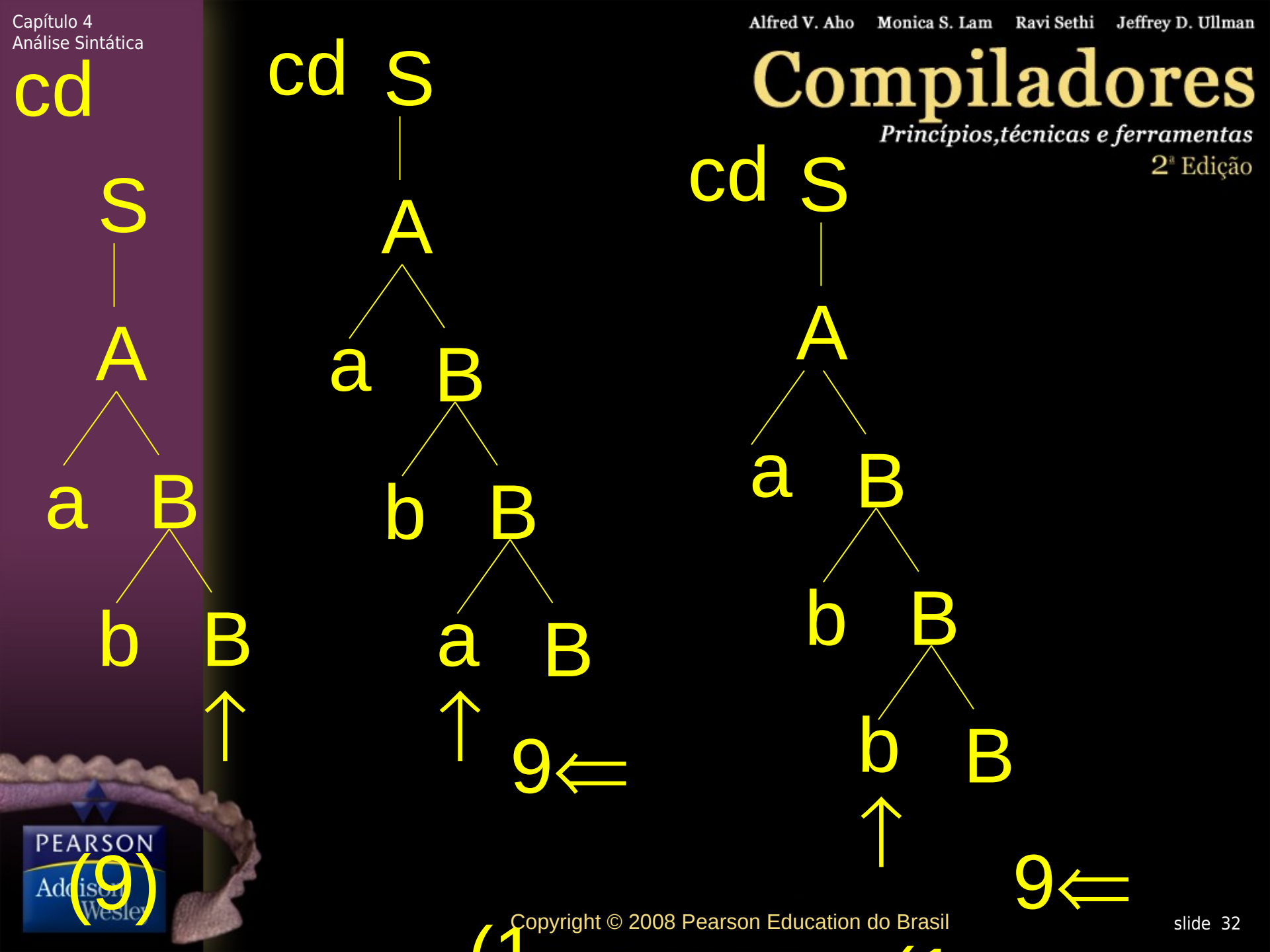
... e a sentença **abcd**, a análise segundo método retrocesso é:

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

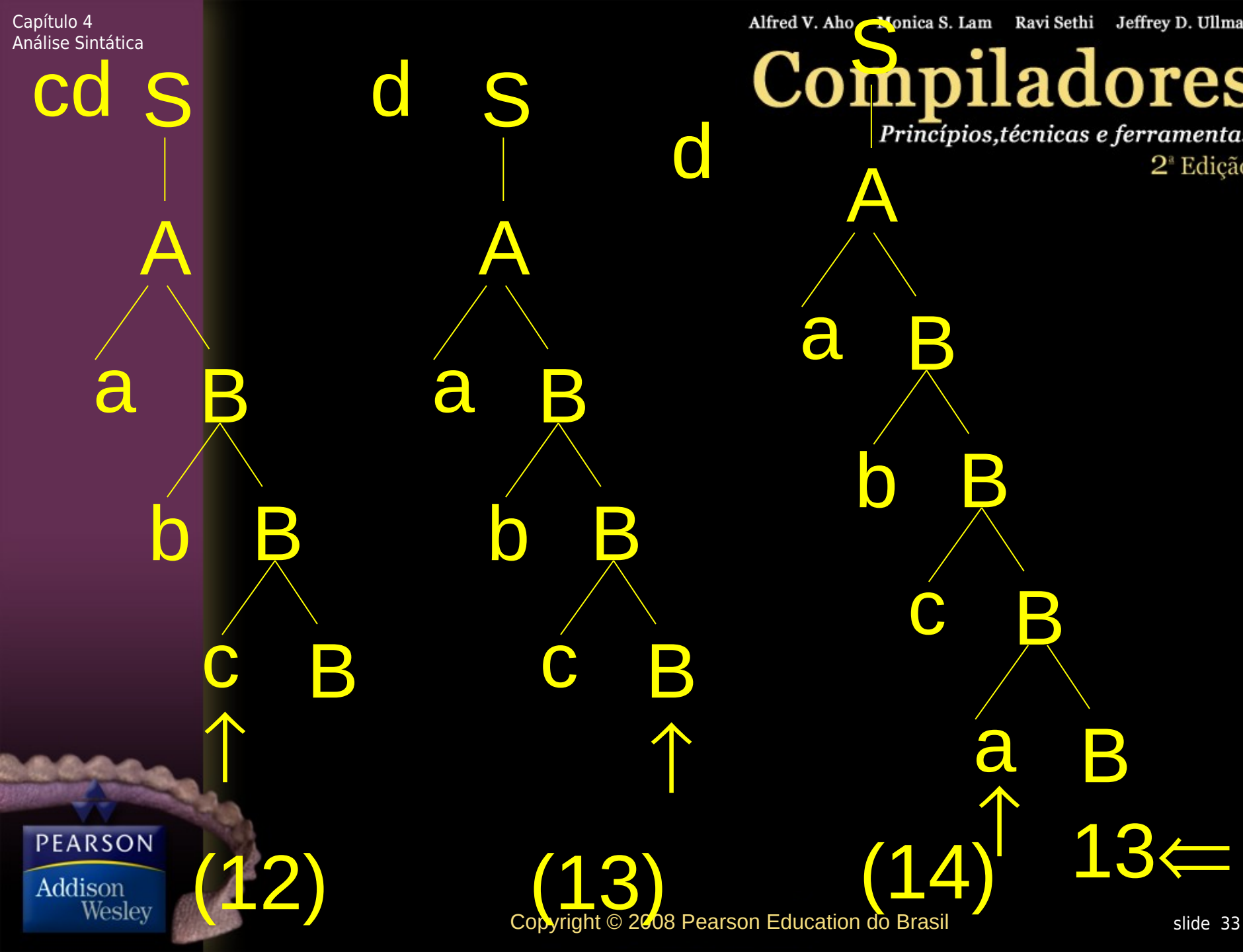
Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição



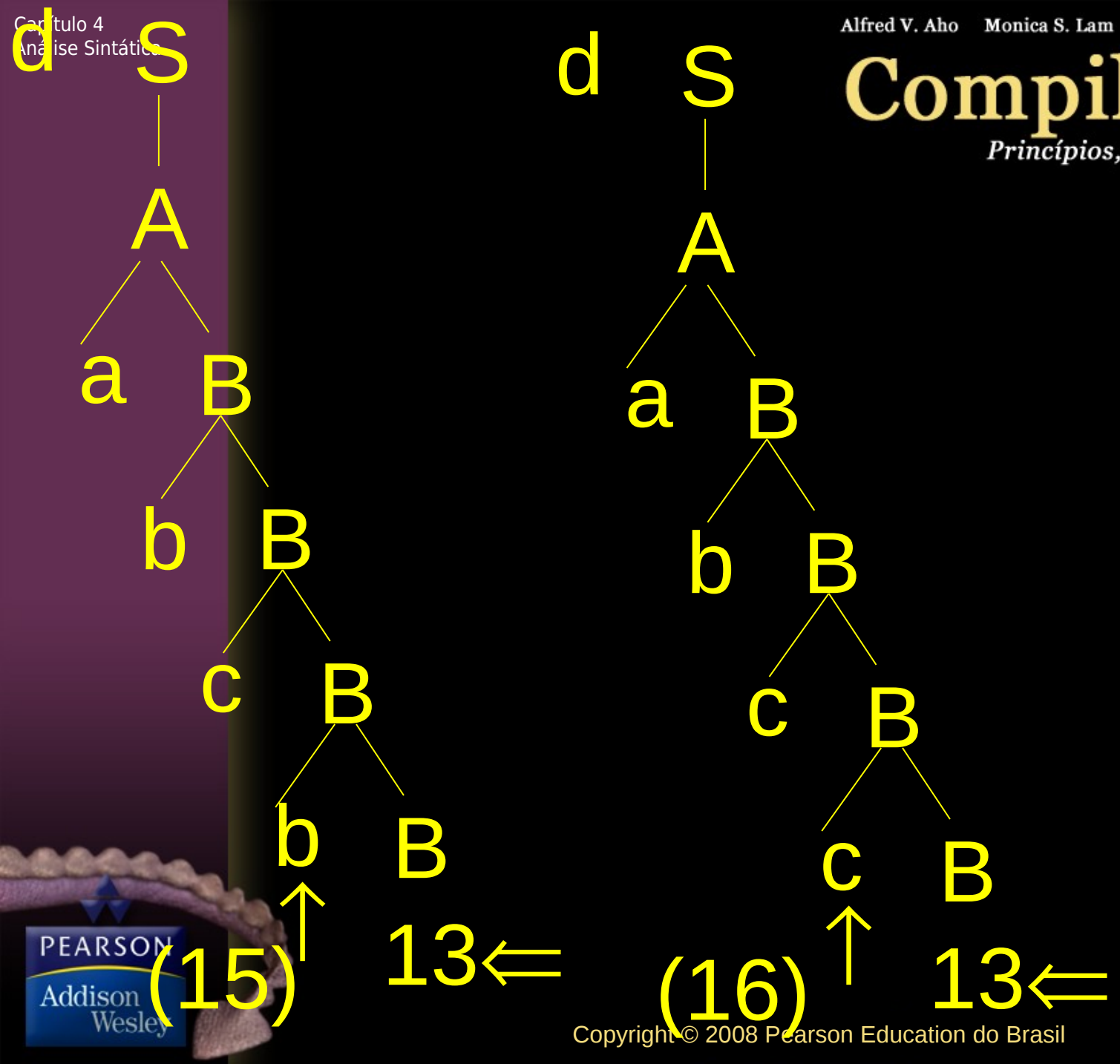
Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição



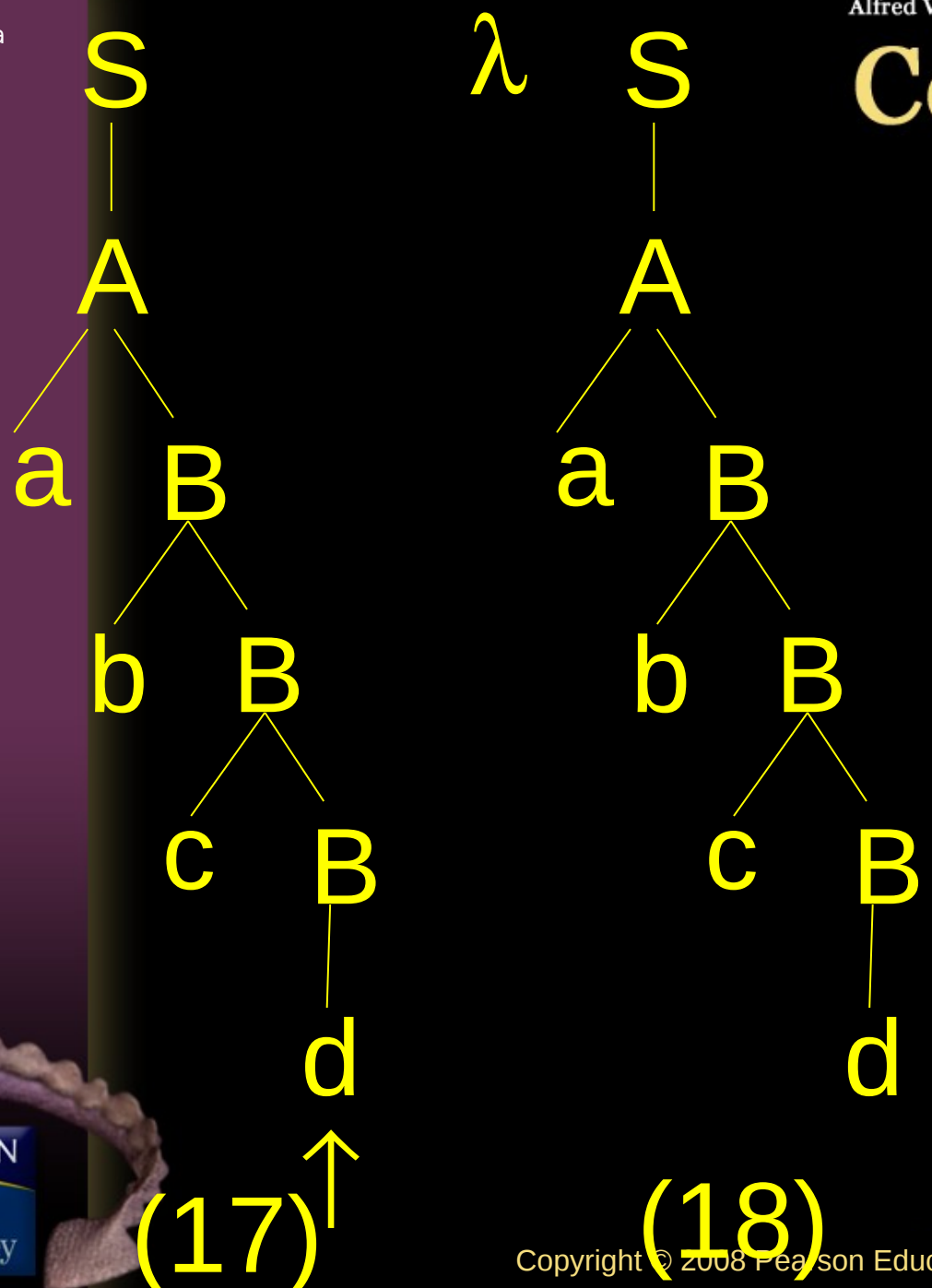
Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Exercício: dada a gramática

G = {Vn, Vt, S, P}

onde Vn={S,A,B,C} Vt = {a,b,c,d}

P: $S \rightarrow AS \mid BA$

$A \rightarrow aB \mid C$

$B \rightarrow bA \mid d$

$C \rightarrow c$

... e a sentença **abcdad**, pede-se a análise segundo método retrocesso.

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Problemas desse método

-eficiência, tempo, necessidade de manter/conhecer a árvore inteira para recuperar estados anteriores
- só se implementa analisador descendente se puder eliminar os retrocessos.

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Método Recursivo Descendente

- dado o símbolo de entrada **a** e o não-terminal A a ser expandido, o método procura qual das produções

$$A \rightarrow \alpha 1 | \alpha 2 | ... |$$

tem seu lado direito um **a**

- se existe $A \rightarrow \lambda$ e nenhuma outra alternativa tem **a**, então aceitar a

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios, técnicas e ferramentas
2ª Edição

Método Recursivo Descendente

- trata-se de um conjunto de procedimentos recursivos, um para cada não-terminal a ser derivado.

restrições ao método:

1. não ter regras com recursão à esquerda (tipo $A \rightarrow A\alpha$)
2. não possuir mais de um lado direito de um não-terminal começando pelo mesmo terminal.

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Exemplo: Seja a gramática G = (Vn, Vt, P, exp)

P = <exp> ::= <termo> + <exp>
<exp> ::= <termo>
<termo> ::= <fator> * <termo>
<termo> ::= <fator>
<fator> ::= <primário> - <fator>
<fator> ::= <primário>
<primário> ::= i
<primário> ::= (<exp>)

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Seja AL uma rotina que traz o próximo item léxico (analisador léxico) na variável símbolo.

```
procedure ANSINT;
  begin
        AL;
        EXP;
  end;
```

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

```
procedure EXP;
  begin
       TERMO;
       if símbolo = '+' then
          begin
               AL;
               EXP;
          end;
  end;
```

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

```
procedure TERMO;
  begin
      FATOR;
      if símbolo = '*' then
         begin
             AL;
             TERMO;
         end;
  end;
```

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

```
procedure FATOR;
  begin
      PRIMÁRIO;
      if símbolo = '-' then
         begin
             AL;
             FATOR;
         end;
  end;
```

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

```
procedure PRIMÁRIO;
  begin
      if símbolo = '(' then
         begin
              AL;
              EXP;
            if símbolo = ')' then AL
                                 else ERRO;
         end
       else if símbolo = 'i' then AL
                                 else ERRO;
  end;
```

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Implementação de Parser Recursivo Descendente

Método: Diagrama Sintático -> Procedimento
Consequência: 1 Gramática -> 1 Programa

Não-terminal -> Grafo -> Procedimento

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Regras para a Construção de Parser Recursivos Descendentes

A - Regras de Transformação

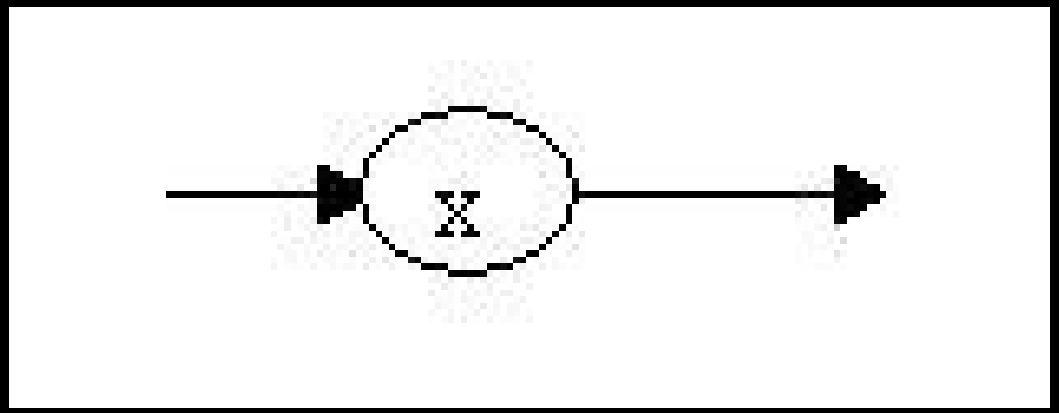
notação BNF para Grafos Sintáticos

1. Cada produção A ::= $\alpha 1 \mid \alpha 2 \mid \ldots \mid \alpha$ é mapeada num grafo de nome A

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Regras para a Construção de Parser Recursivos Descendentes

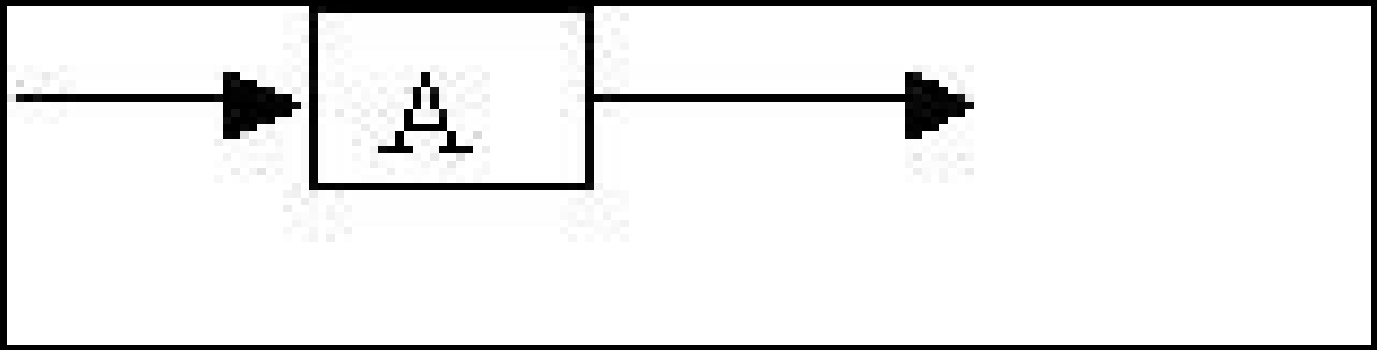
2. Toda ocorrência de um terminal x numa forma α corresponde ao seu reconhecimento na cadeia de entrada e a leitura do próximo símbolo dessa cadeia.



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Regras para a Construção de Parser Recursivos Descendentes

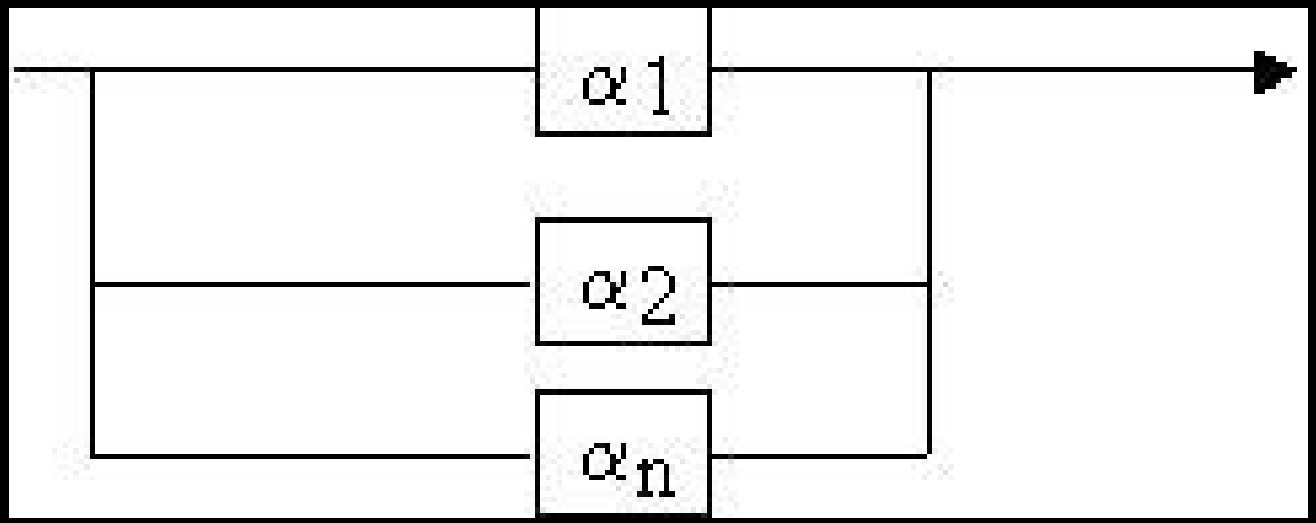
3. Toda ocorrência de um não-terminal A num α corresponde a análise imediata de A.



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Regras para a Construção de Parser Recursivos Descendentes

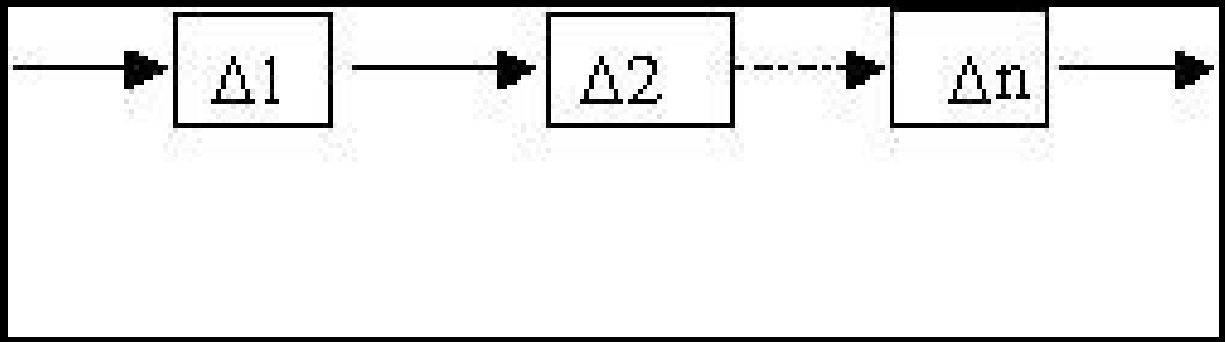
4. Alternativas são representadas como :



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Regras para a Construção de Parser Recursivos Descendentes

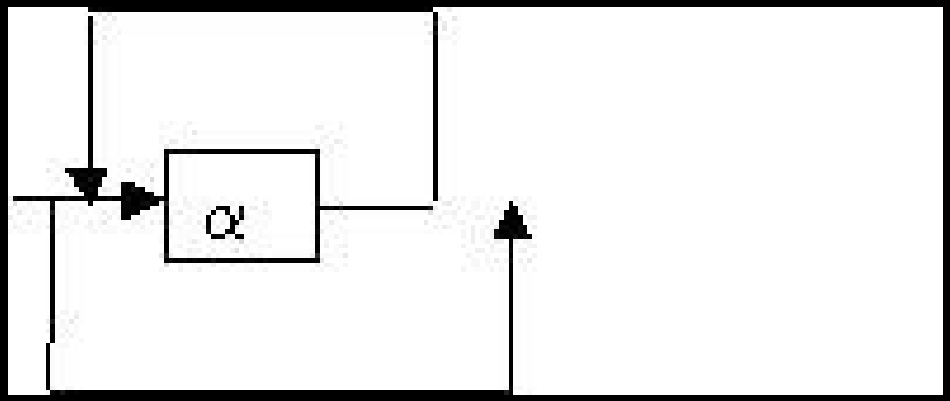
5. Um α do tipo $\Delta 1\ \Delta 2 \ldots \Delta n$ é mapeado em:



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Regras para a Construção de Parser Recursivos Descendentes

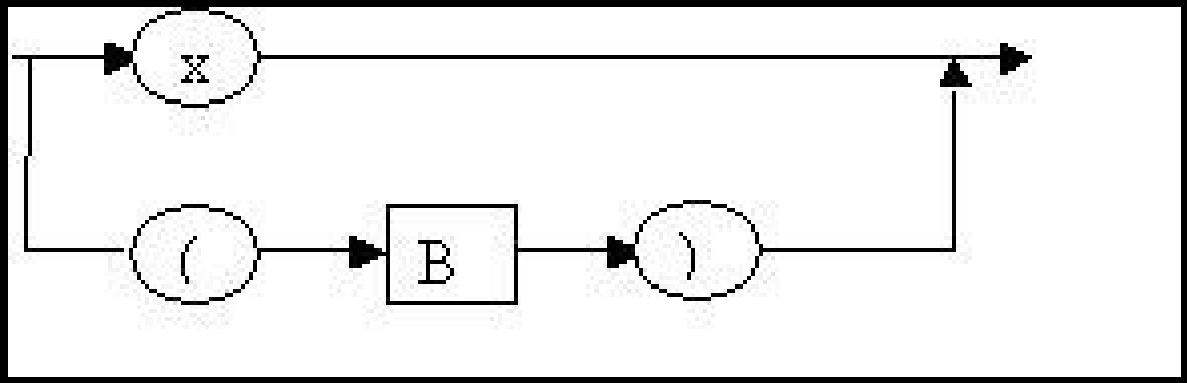
6. Um Δ da forma $\{\alpha\}^*$ ou α^* é representado por:



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

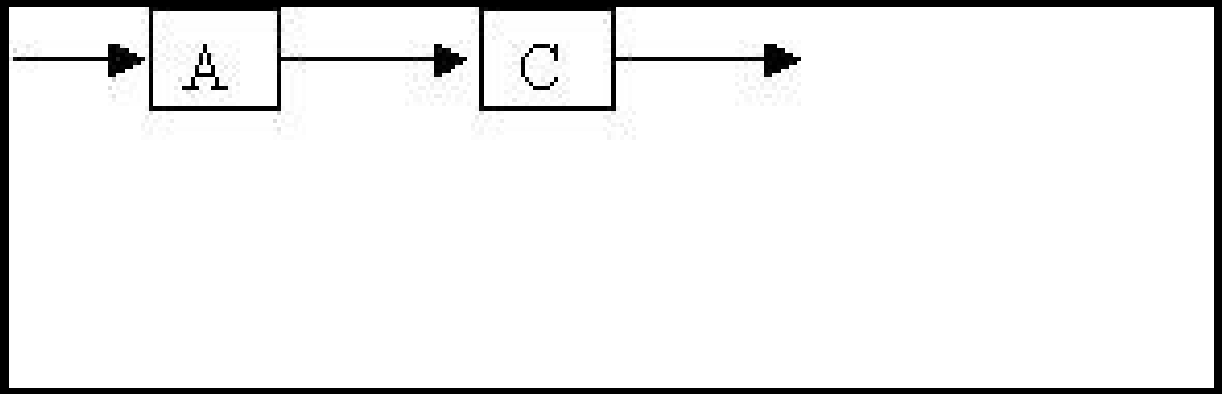
Exemplo: A –> x | (B)
B –> AC
C –> +AC | λ

A :



Exemplo: A –> x | (B)
B –> AC
C –> +AC | λ

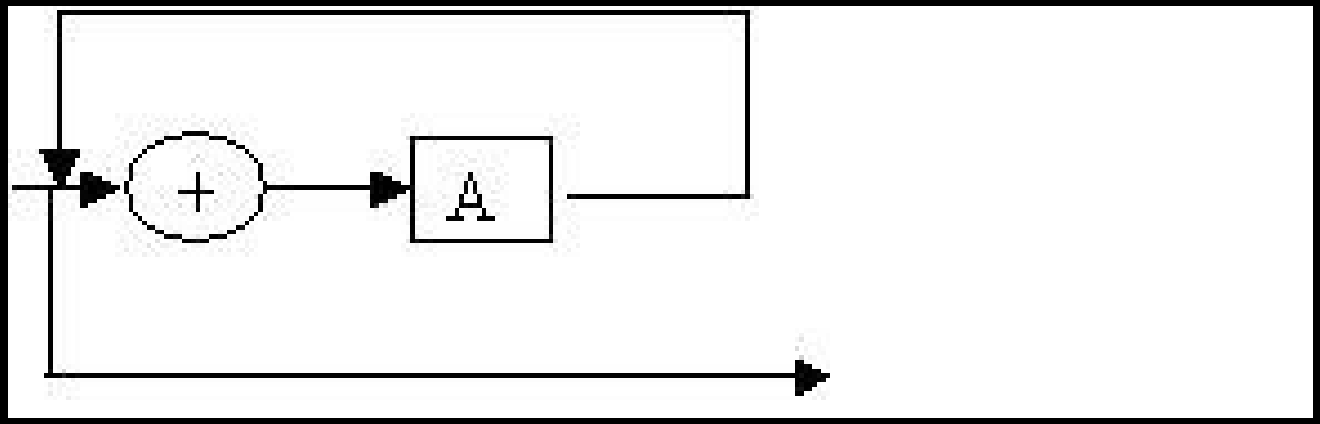
B :



Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Exemplo: A –> x | (B)
B –> AC
C –> +AC | λ

C :



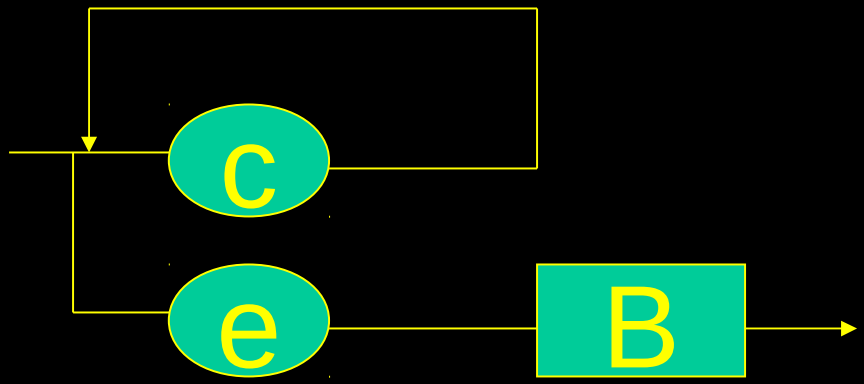
Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios técnicas e ferramentas
2ª Edição

Exemplo: grafos sintáticos + algoritmo

$S \rightarrow aAd \quad A \rightarrow cA \mid eB \quad B \rightarrow f \mid g$

S:

A:



B:

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

```
procedure S;
    begin
        if ch = 'a' then
            begin
                atualiza ch;
                proxsimb;
                A;
                if ch = 'd' then proxsimb
                else ERRO
            end
        else ERRO
    end
```

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

```
procedure A;
    begin
        case ch of
            'c': begin
                    proxsimb;
                    A;
                 end;
            'e': begin
                    proxsimb;
                    B;
                 end;
            else ERRO
        end
end
```

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

```
procedure B;
    begin
        if ch = 'f' or ch = 'g' then proxsimb
        else ERRO
    end;

    /*Programa principal*/
    begin
        proxsimb;
        S;
        if terminou_cadeia then SUCESSO
        else INSUCESSO
    end.
```

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Exerc1: fazer os grafos sintáticos para a gramática de expressões vista.

Exerc2: fazer os grafos sintáticos e o algoritmo o analisador descendente da gramática abaixo

S ::= AS | BA
A ::= aB | C
B ::= bA | d
C ::= c

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Exerc3: faça os procedimentos e os grafos sintáticos para a gramática

<Cmdos> ::= <Cmdo> ; <Cmdos> | <Cmdo>
<Cmdo> ::= if exp then <Cmdo> <Pelse>
| for id := exp to exp do <Cmdo>
| while exp do <Cmdo>
| id := exp
| begin <Cmdos> end
<Pelse> ::= else <Cmdo> | ε

Alfred V. Aho Monica S. Lam Ravi Sethi Jeffrey D. Ullman
Compiladores
Princípios,técnicas e ferramentas
2ª Edição

Verifique se a sentença é válida para a gramática

x := exp; y := exp
w := if exp then if exp then z := exp
else k := exp