

Algoritmo de Early

♦ Early

- desenvolvido em 1968
- considerado o **mais rápido algoritmo** de reconhecimento conhecido para GLC
- **tempo** de processamento
 - * **proporcional** a $|w|^3$
 - * para gramáticas **não-ambíguas**, pode ser **proporcional** a $|w|^2$
 - * para muitas **gramáticas de interesse prático**, o tempo é **proporcional** a $|w|$

♦ Algoritmo

- tipo *top-down*
- a partir de uma GLC qualquer executa sempre a **derivação mais à esquerda**
- **cada ciclo gera um terminal**
 - * comparado com o símbolo da entrada
- comparação com sucesso

* construção de um conjunto de produções que pode gerar o próximo símbolo

♦ Algoritmo de Early

- $G = (V, T, P, S)$ uma GLC qualquer
- $w = a_1a_2...a_n$ palavra a ser reconhecida
- símbolo "."
 - * é usado como um marcador
 - * em cada produção antecede a posição que será analisada na tentativa de gerar o próximo terminal
- sufixo "/u"
 - * adicionado a cada produção
 - * indica o u-ésimo ciclo em que esta produção passou a ser considerada

a) *Construção do primeiro conjunto de produções*

- produções que partem de S
- produções que podem ser aplicadas em sucessivas derivações mais à esquerda a partir de S

$D_0 = \emptyset$

para toda produção $S \rightarrow \alpha \in P$ (1)

faça $D_0 = D_0 \cup \{S \rightarrow \cdot \alpha / 0\}$

repita (2)

para toda produção $A \rightarrow \cdot B\beta / 0 \in D_0$

faça

para toda produção $B \rightarrow \phi \in P$

faça $D_0 = D_0 \cup \{B \rightarrow \cdot \phi / 0\}$

até não ocorrerem mais inclusões

(1) produções que partem de S

(2) as produções que podem ser aplicadas em derivação mais à esquerda a partir de S

b) *Construção dos demais conjuntos de produção*

- $n = |w|$ conjuntos de produção a partir de D_0
- ao gerar o símbolo a_r de w constrói D_r : produções que podem gerar o símbolo a_{r+1}

```
para r variando de 1 até n // cada ciclo gera um conjunto de produções  $D_r$ 
faça  $D_r = \emptyset$ ;
para toda  $A \rightarrow \alpha.a_r\beta/s \in D_{r-1}$  // gera símbolo  $a_r$ 
faça  $D_r = D_r \cup \{A \rightarrow \alpha a_r.\beta/s\}$ ;
repita
    para toda  $A \rightarrow \alpha.B\beta/s \in D_r$  (3)
    faça
        para toda  $B \rightarrow \phi \in P$ 
        faça  $D_r = D_r \cup \{B \rightarrow .\phi/r\}$ 
    para toda  $A \rightarrow \alpha./s$  de  $D_r$  (4)
    faça para toda  $B \rightarrow \beta.A\phi/k \in D_s$ 
        faça  $D_r = D_r \cup \{B \rightarrow \beta A.\phi/k\}$ 
até não ocorrerem mais inclusões
```

(3) produções que podem derivar o próximo símbolo

(4) uma subpalavra de w foi reduzida à variável A : inclui em D_r produções de que referenciam $.A$

c) *Condição de aceitação da entrada*

- se uma produção $S \rightarrow \alpha./0$ pertence a D_n , então a palavra w de entrada foi aceita
- $S \rightarrow \alpha./0$ é uma produção que
 - * parte do símbolo inicial S
 - * foi incluída em D_0 (" $/0$ ")
 - * todo o lado direito da produção foi analisado com sucesso (" $."$ está no final de α)

♦ **Otimização das etapas a) e b)**

- ciclos *repita-até*
 - * restritos exclusivamente às produções recentemente incluídas em D_r ou em D_0 ainda não-analisadas

♦ *Exemplo. Gramática análoga à definição de "expressão simples" do PASCAL*

$G = (\{E, T, F\}, \{+, *, [,], x\}, P, E)$, onde

$P = \{E \rightarrow T \mid E+T, T \rightarrow F \mid T*F, F \rightarrow (E) \mid x\}$

reconhecimento da palavra $x*x$

D_0 :

$E \rightarrow .T/0$ produções que partem

$E \rightarrow .E+T/0$ do símbolo inicial

$T \rightarrow .F/0$ produções que podem ser aplicadas

$T \rightarrow .T*F/0$ em derivação mais à esquerda

$F \rightarrow .(E)/0$ a partir do símbolo inicial

$F \rightarrow .x/0$

D_1 : reconhecimento de " x " em $\underline{x}*x$

$F \rightarrow x./0$ x foi reduzido à F

$T \rightarrow F./0$ inclui as produções de D_0 ($F \rightarrow x./0$)

$T \rightarrow T.*F/0$ que referenciam $.F$ direta ou

$E \rightarrow T./0$ indiretamente, movendo "."

$E \rightarrow E.+T/0$ um símbolo para a direita

D_2 : reconhecimento de "*" em $x*\underline{x}$

$T \rightarrow T*.F/0$ gerou *; o próximo será gerado por F

$F \rightarrow .(E)/2$ inclui as produções de P que podem

$F \rightarrow .x/2$ gerar o próximo terminal a partir de F

D_3 : reconhecimento de " x " em $x*\underline{x}$

$F \rightarrow x./2$ x foi reduzido à F

$T \rightarrow T*F./0$ incluído de D_2 ($F \rightarrow x./2$);
a entrada foi reduzida à T ;

$E \rightarrow T./0$	incluído de $D_0 (T \rightarrow T*F./0)$; a entrada foi reduzida à E ;
$T \rightarrow T.*F/0$	incluído de $D_0 (T \rightarrow T*F./0)$;
$E \rightarrow E.+T/0$	incluído de $D_0 (E \rightarrow T./0)$.

$w = x*x$ foi reduzida ao símbolo inicial E

- $E \rightarrow T./0$ pertence a D_3
- a entrada foi aceita