

[Aula 17] Reconhecimento das LLC – Algoritmo de Earley

Prof. João F. Mari
joaof.mari@ufv.br

BIBLIOGRAFIA

- MENEZES, P. B. **Linguagens formais e autômatos**, 6. ed., Bookman, 2011.
 - Capítulo 7.
 - + Slides disponibilizados pelo autor do livro.



ROTEIRO

- Algoritmo de Earley
- **DEFINIÇÃO:** Algoritmo de Earley
- **EXEMPLO:** Algoritmo de Earley

Algoritmo de Earley

- Algoritmo de Earley (1968)
 - Possivelmente o mais rápido algoritmo para reconhecimento de LLC;
 - Tempo de processamento proporcional a:
 - Em geral: $|w|^3$
 - Gramáticas não-ambíguas: $|w|^2$
 - Muitas gramáticas de interesse prático: $|w|$

AN EFFICIENT CONTEXT-FREE PARSING ALGORITHM

Jay Earley

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania
August, 1968

Submitted to Carnegie-Mellon University
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Algoritmo de Earley

- Ideia do algoritmo:
 - Algoritmo *top-down*;
 - Parte de uma GLC sem produções vazias:
 - A partir do símbolo inicial:
 - Executa sempre a derivação mais à esquerda;
 - Cada ciclo gera um terminal
 - Comparado com o símbolo da entrada
 - **SUCESSO**: construção do conjunto de produções que, potencialmente, pode gerar o próximo símbolo.

DEFINIÇÃO: Algoritmo de Earley

- Seja G uma GLC sem produções vazias:
 - $G = (V, T, P, S)$
- E w é a palavra a ser verificada:
 - $w = a_1a_2\dots a_n$ palavra a ser verificada.
- O marcador “•”:
 - Antecedendo a posição, em cada produção, que será analisada;
 - na tentativa de gerar o próximo símbolo terminal.
- O sufixo “/u”:
 - Adicionado a cada produção;
 - O u -ésimo ciclo em que passou a ser considerada.

DEFINIÇÃO: Algoritmo de Earley

- *Etapa 1: construção de D_0 : primeiro conjunto de produções*
 - Produções que partem de S (1)
 - Produções que podem ser aplicadas (2)
 - Em sucessivas derivações mais à esquerda (a partir de S)
1. $D_0 = \emptyset$
 2. **para** toda $S \rightarrow \alpha \in P$ **faça** (1)
 3. $D_0 = D_0 \cup \{ S \rightarrow \bullet\alpha/0 \}$
 4. **repita para** toda $A \rightarrow \bullet B\beta/0 \in D_0$ **faça** (2)
 5. **para** toda $B \rightarrow \varphi \in P$ **faça**
 6. $D_0 = D_0 \cup \{ B \rightarrow \bullet\varphi/0 \}$
 7. **até** que o cardinal de D_0 não aumente

DEFINIÇÃO: Algoritmo de Earley

- *Etapa 2: construção dos demais conjuntos de produção:*
 - $n = |w|$ conjuntos de produção a partir de D_0
 - ao gerar a_r , constrói D_r : produções que podem gerar a_{r+1}
1. **para** r variando de 1 até n **faça** (1)
 2. $D_r = \emptyset$;
 3. **para** toda $A \rightarrow \alpha \bullet a_r \beta / s \in D_{r-1}$ **faça** (2)
 4. $D_r = D_r \cup \{ A \rightarrow \alpha a_r \bullet \beta / s \}$;
 5. **repita**
 6. **para** toda $A \rightarrow \alpha \bullet B \beta / s \in D_r$ **faça** (3)
 7. **para** toda $B \rightarrow \varphi \in P$ **faça**
 8. $D_r = D_r \cup \{ B \rightarrow \bullet \varphi / r \}$
 9. **para** toda $A \rightarrow \alpha \bullet / s$ de D_r **faça** (4)
 10. **para** toda $B \rightarrow \beta \bullet A \varphi / k \in D_s$ **faça**
 11. $D_r = D_r \cup \{ B \rightarrow \beta A \bullet \varphi / k \}$
 12. **até** que o cardinal de D_r não aumente

DEFINIÇÃO: Algoritmo de Earley

- *Etapa 3: condição de aceitação da entrada:*
 - A palavra w é aceita se:
 - Uma produção da forma $S \rightarrow \alpha \bullet / 0$ pertence a D_n
 - $S \rightarrow \alpha \bullet / 0$ 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:
 - (" \bullet " está no final de α).
- Otimização do Algoritmo de Earley:
 - Os ciclos repita-até:
 - Podem ser restritos exclusivamente às produções recentemente incluídas em D_r ou em D_0 ainda não-analisadas.

EXEMPLO: Algoritmo de Earley

- "Expressão simples" da linguagem Pascal
 - $G = (\{ E, T, F \}, \{ +, *, [,], x \}, P, E)$, na qual:
 - $P = \{ E \rightarrow T \mid E+T,$
 - $T \rightarrow F \mid T * F,$
 - $F \rightarrow [E] \mid x \}$
- Reconhecimento da palavra $w = x*x$:
 - $|w| = n = 3$

EXEMPLO: Algoritmo de Earley

• Reconhecimento da palavra $x*x$:

– D_0 (Etapa 1):

- $E \rightarrow \bullet T/0$ (1) Produções que partem do símbolo inicial
- $E \rightarrow \bullet E+T/0$
- $T \rightarrow \bullet F/0$ (2) Produções que podem ser aplicadas em derivação mais à esquerda a partir do símbolo inicial
- $T \rightarrow \bullet T*F/0$
- $F \rightarrow \bullet [E]/0$
- $F \rightarrow \bullet x/0$

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

1. $D_0 = \emptyset$
2. **para** toda $S \rightarrow \alpha \in P$ **faça** (1)
3. $D_0 = D_0 \cup \{ S \rightarrow \bullet \alpha / 0 \}$
4. **repita para** toda $A \rightarrow \bullet B\beta / 0 \in D_0$ **faça** (2)
5. **para** toda $B \rightarrow \varphi \in P$ **faça**
6. $D_0 = D_0 \cup \{ B \rightarrow \bullet \varphi / 0 \}$
7. **até** que o cardinal de D_0 não aumente

EXEMPLO: Algoritmo de Earley

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

- $F \rightarrow x\bullet/0$ # x foi reduzido a F
- $0''$ – $T \rightarrow F\bullet/0$ # inclui todas as produções de D_0 que referenciaram $\bullet F$ direta ou indiretamente
- $T \rightarrow T\bullet*F/0$ #
- $1''$ – $E \rightarrow T\bullet/0$ # movendo o marcador " \bullet "
- $2''$ – $E \rightarrow E\bullet+T/0$ # um símbolo para a direita

• D_0 :

- $E \rightarrow \bullet T/0$
- $E \rightarrow \bullet E+T/0$
- $T \rightarrow \bullet F/0$
- $T \rightarrow \bullet T*F/0$
- $F \rightarrow \bullet [E]/0$
- $F \rightarrow \bullet x/0$

- para** r variando de 1 até n **faça** (1)
- $D_r = \emptyset;$
- para** toda $A \rightarrow \alpha \bullet a_r \beta / s \in D_{r-1}$ **faça** (2)
- $D_r = D_r \cup \{ A \rightarrow \alpha a_r \bullet \beta / s \};$
- repita # 0, 1, 2**
- para** toda $A \rightarrow \alpha \bullet B\beta / s \in D_r$ **faça** # ' (3)
- para** toda $B \rightarrow \varphi \in P$ **faça**
- $D_r = D_r \cup \{ B \rightarrow \bullet \varphi / r \}$
- para** toda $A \rightarrow \alpha \bullet / s$ de D_r **faça** # " (4)
- para** toda $B \rightarrow \beta \bullet A\varphi / k \in D_s$ **faça**
- $D_r = D_r \cup \{ B \rightarrow \beta A \bullet \varphi / k \}$
- até** que o cardinal de D_r não aumente

EXEMPLO: Algoritmo de Earley

- D_2 : reconhecimento de $*$ em $x*x$
 - $T \rightarrow T* \bullet F/0$ # gerou $*$; o próximo será gerado por F
 - $F \rightarrow \bullet [E]/2$ # inclui todas as produções de P que
 - $0'$ – $F \rightarrow \bullet x/2$ # podem gerar o próximo terminal a partir de F

- D_1 :
 - $F \rightarrow x \bullet /0$
 - $T \rightarrow F \bullet /0$
 - $T \rightarrow T \bullet * F/0$
 - $E \rightarrow T \bullet /0$
 - $E \rightarrow E \bullet + T/0$

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

$P = \{E \rightarrow T \mid E+T,$
 $T \rightarrow F \mid T*F,$
 $F \rightarrow [E] \mid x \}$

```

para r variando de 1 até n faça (1)
   $D_r = \emptyset$ ;
  para toda  $A \rightarrow \alpha \bullet a_r \beta / s \in D_{r-1}$  faça (2)
     $D_r = D_r \cup \{ A \rightarrow \alpha a_r \bullet \beta / s \}$ ;
  repita # 0
    para toda  $A \rightarrow \alpha \bullet B \beta / s \in D_r$  faça # ' (3)
      para toda  $B \rightarrow \varphi \in P$  faça
         $D_r = D_r \cup \{ B \rightarrow \bullet \varphi / r \}$ 
      para toda  $A \rightarrow \alpha \bullet / s$  de  $D_r$  faça # " (4)
        para toda  $B \rightarrow \beta \bullet A \varphi / k \in D_s$  faça
           $D_r = D_r \cup \{ B \rightarrow \beta A \bullet \varphi / k \}$ 
    até que o cardinal de  $D_r$  não aumente
  
```

EXEMPLO: Algoritmo de Earley

- D_3 : reconhecimento de x em $x*x$
 - $F \rightarrow x \bullet /2$ # x foi reduzido à F
 - $0''$ – $T \rightarrow T*F \bullet /0$ # incluído de D_2 ($F \rightarrow x \bullet /2$); entrada reduzida à T
 - $E \rightarrow T \bullet /0$ # incluído de D_0 ($T \rightarrow T*F \bullet /0$); entrada reduzida à E
 - $1''$ – $T \rightarrow T \bullet * F/0$ # incluído de D_0 (pois $T \rightarrow T*F \bullet /0$)
 - $2''$ – $E \rightarrow E \bullet + T/0$ # incluído de D_0 (pois $E \rightarrow T \bullet /0$)

- $D_{r-1} = D_2$:
 - $T \rightarrow T \bullet * F/0$
 - $F \rightarrow \bullet [E]/2$
 - $F \rightarrow \bullet x/2$

- D_0 :
 - $E \rightarrow \bullet T/0$
 - $E \rightarrow \bullet E+T/0$
 - $T \rightarrow \bullet F/0$
 - $T \rightarrow \bullet T*F/0$
 - $F \rightarrow \bullet [E]/0$
 - $F \rightarrow \bullet x/0$

```

para r variando de 1 até n faça (1)
   $D_r = \emptyset$ ;
  para toda  $A \rightarrow \alpha \bullet a_r \beta / s \in D_{r-1}$  faça (2)
     $D_r = D_r \cup \{ A \rightarrow \alpha a_r \bullet \beta / s \}$ ;
  repita # 0, 1, 2
    para toda  $A \rightarrow \alpha \bullet B \beta / s \in D_r$  faça # ' (3)
      para toda  $B \rightarrow \varphi \in P$  faça
         $D_r = D_r \cup \{ B \rightarrow \bullet \varphi / r \}$ 
      para toda  $A \rightarrow \alpha \bullet / s$  de  $D_r$  faça # " (4)
        para toda  $B \rightarrow \beta \bullet A \varphi / k \in D_s$  faça
           $D_r = D_r \cup \{ B \rightarrow \beta A \bullet \varphi / k \}$ 
    até que o cardinal de  $D_r$  não aumente
  
```

EXEMPLO: Algoritmo de Earley

- A entrada w é aceita:
 - Pois $w = x*x$ foi reduzida a E e;
 - $E \rightarrow T\bullet/0$ pertence a D_3 .

[FIM]

- FIM:
 - **[AULA 17]** Propriedades e reconhecimento das LLC – Algoritmo de Earley
- Próxima aula:
 - **[AULA 18]** Linguagens recursivamente enumeráveis – Máquina de Turing