

SCC0217 – ASD preditiva recursiva

Guilherme de Abreu Barreto, 12543033

Hélio Nogueira Cardoso, 10310227

Theo da Mota dos Santos, 10691331

Laura Fernandes Camargos, 13692334

Sandy da Costa Dutra, 12544570

12 de abril de 2025

1 Exercício

Considerando a seguinte gramática livre de contexto abaixo, é preciso determinar se ela é LL(1), construir o grafo de análise sintática e os procedimentos equivalentes.

1.1 Gramática

A gramática analisada é:

$$\begin{aligned} \langle A \rangle &::= x \mid (\langle B \rangle) \\ \langle B \rangle &::= \langle A \rangle \langle C \rangle \\ \langle C \rangle &::= + \langle A \rangle \langle C \rangle \mid \lambda \end{aligned}$$

Primeiramente, observamos que não existe nenhum tipo de recursão à esquerda direta ou indireta na gramática. Além disso, observamos que em uma mesma linha, não temos conjuntos FIRST iguais entre os possíveis termos derivacionais que possam resultar em confusão no analisador léxico. Assim, a gramática acima pode ser dita como LL(1).

Agora, para a montagem do grafo, teremos:

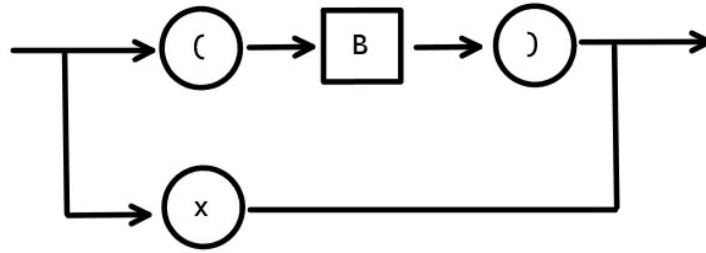


Figura 1: Grafo do termo não terminal A

Algorithm 1 Procedimento de análise sintática para o não terminal A

```

1: procedimento A
2: begin
3: if símbolo = 'x' then
4:   obter_símbolo
5: else if símbolo = '(' then
6:   obter_símbolo
7:   B
8:   if símbolo = ')' then
9:     obter_símbolo
10:  else
11:    ERRO
12:  end if
13: else
14:   ERRO
15: end if
16: end
  
```

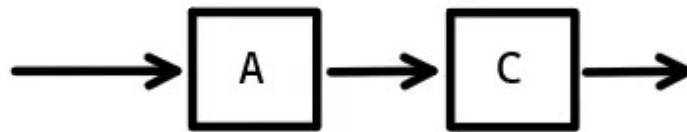


Figura 2: Grafo do termo não terminal B

Algorithm 2 Procedimento de análise sintática para o não terminal B

```

1: procedimento B
2: begin
3: A
4: C
5: end
  
```

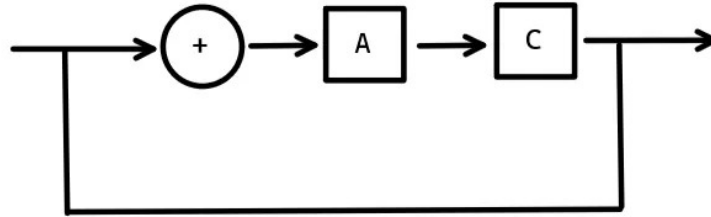


Figura 3: Grafo do termo não terminal C

Algorithm 3 Procedimento de análise sintática para o não terminal C

```

1: procedimento C
2: begin
3: if símbolo = '+' then
4:   obter_símbolo
5:   A
6:   C
7: end if
8: end

```

Algorithm 4 Procedimento de análise sintática para o programa completo

```

1: procedimento ASD1
2: begin
3: obter_símbolo
4: A
5: if terminou_cadeia then
6:   SUCESSO
7: else
8:   ERRO
9: end if
10: end

```

Algorithm 5 Procedimento de análise sintática para o programa completo

```

1: procedimento ASD2
2: begin
3: obter_símbolo
4: if procedimentoA() and terminou_cadeia then
5:   SUCESSO
6: else
7:   ERRO
8: end if
9: end

```
