

Compiladores

Universidade Federal do Oeste do Pará

Maio 03, 2023

Santarém, Pará - Brasil



Trabalho 1

Equipe 05

Fellype Siqueira Barroso, Lucas Darlindo Freitas Rodrigues

{fellypecsiqueira, lucas.darlindo}@gmail.com

Sobre o código: Para acessar o repositório contendo o código-fonte com este documento e adicionais, além do Parser desenvolvido para a Gramática LL(1) após adaptações, clique aqui.

- Dada a Gramática 5, com as seguintes regras:

```
ASSIGN ::= LEFT > (EXPR) | LEFT < (EXPR) | LEFT = REST
LEFT  ::= ID | ID[EXPR]
REST  ::= LEFT = REST | (EXPR)
EXPR  ::= EXPR + TERM | EXPR - TERM | TERM
TERM  ::= TERM * UNARY | TERM / UNARY | UNARY
UNARY ::= + UNARY | - UNARY | FACTOR
FACTOR ::= (EXPR) | DIGIT | LEFT
ID     ::= a | b | ... | z
DIGIT  ::= 0 | 1 | ... | 9
```

1. Ajustar a gramática para torná-la LL(1)

```
ASSIGN ::= LEFT ASSIGN'
ASSIGN' ::= > (EXPR) | < (EXPR) | = REST
LEFT  ::= ID LEFT'
LEFT' ::= [EXPR] | ε
REST  ::= LEFT = REST | (EXPR)
EXPR  ::= TERM EXPR'
EXPR' ::= + TERM EXPR' | - TERM EXPR' | ε
TERM  ::= UNARY TERM'
TERM' ::= * UNARY TERM' | / UNARY TERM' | ε
UNARY ::= + UNARY | - UNARY | FACTOR
```

$\text{FACTOR} ::= (\text{EXPR}) \mid \text{DIGIT} \mid \text{LEFT}$
 $\text{ID} ::= a \mid b \mid \dots \mid z$
 $\text{DIGIT} ::= 0 \mid 1 \mid \dots \mid 9$

2. Apresentar 5 *strings* que são aceitas pela gramática, sendo que juntas elas devem utilizar todas as regras da gramática. E em seguida mostrar a árvore de derivação para cada uma dessas *strings*.

- (a) Atribuição a uma variável escalar com operador $<$

$a < ((a + b) * (c - d))$

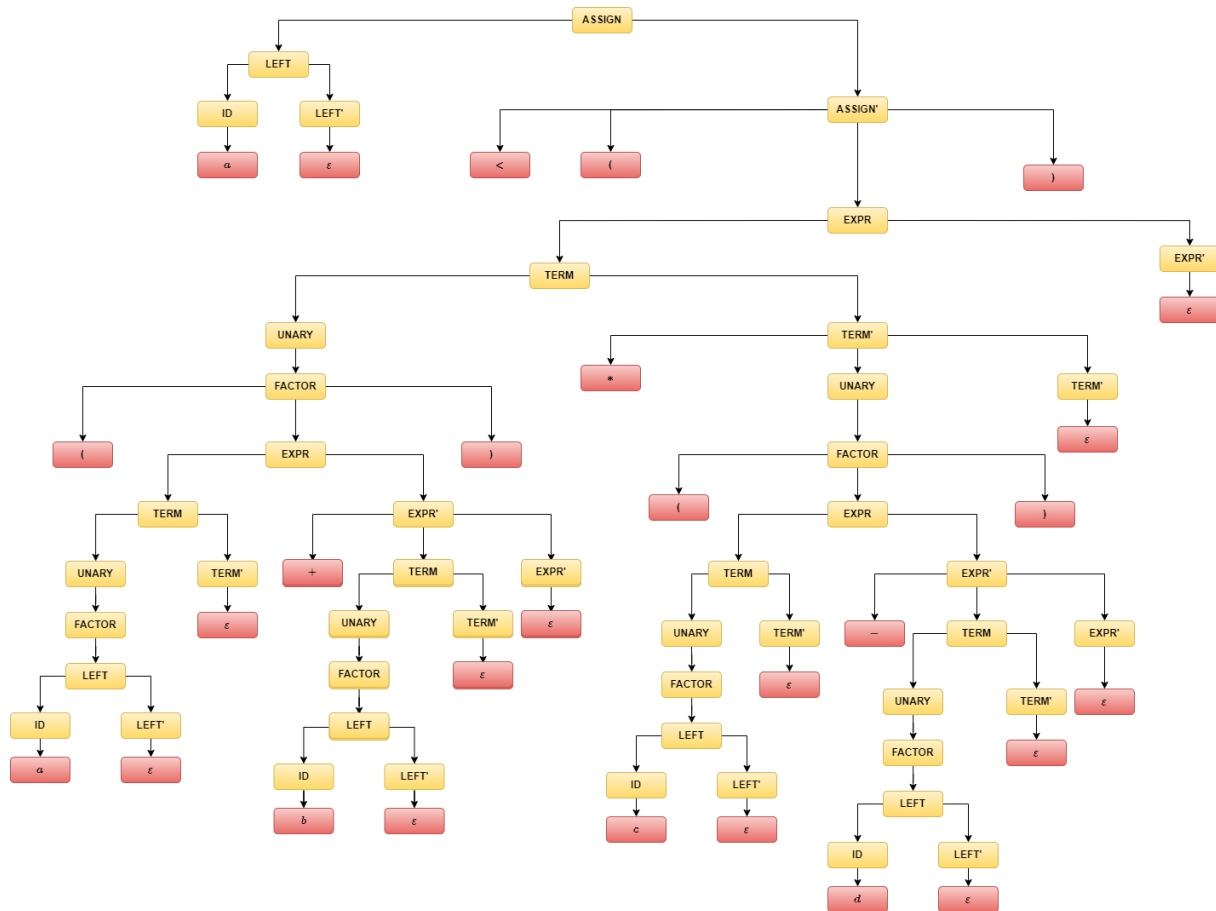
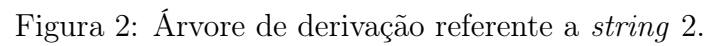


Figura 1: Árvore de derivação referente a *string* 1.

- (b) Atribuição a uma variável escalar com operador $>$

$b > (5 + a - b * (c / 8))$



```
d = ((a / b) * (c * 9) * a[y])
```



