Aula 3 – Exercício sobre descrição de sintaxe

Aluno: Gustavo Camerino de Carvalho
RGA: 202211722035

---

1) Resposta

Gramática para números inteiros

```
<Num> → <Num><digit> | <digit>
<digit> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Derivação mais à direita para o inteiro 4520:

```
<Num>
=> <Num> <digit>
=> <Num> 0
=> <Num> 2 0
=> <Num> 5 2 0
=> 4 5 2 0
```

---

2) Resposta

Gramática para dar precedência ao + sobre * à direita

```
<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <term> + <expr> | <term>
<term> → <factor> * <term> | <factor>
<factor> → (<expr>) | <id>
```

---

3) Resposta

Adicionando os operadores unários ++ e - -

```
<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <term> + <expr> | <term>
<term> → <factor> * <term> | <factor>
<factor> → (<expr>) | <id> | ++ <id> | -- <id>
```

---

4) Resposta

Derivação de expressões

```
<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <id> + <expr> | <id> * <expr> | (<expr>) | <id>
```

(A) A = (B + (C * A))

```
A = <expr>
A = (<id> + <expr>)
A = (<id> + (<expr>))
A = (<id> + (<id> * <expr>))
A = (<id> + (<id> * <id>))
A = (B + (C * A))
```

(B) A = B + C + A

```
A = <expr>
A = <id> + <expr>
A = <id> + <id> + <expr>
A = <id> + <id> + <id>
A = B + C + A
```

(C) A = A * (B + C)

```
A = <expr>
A = <id> * <expr>
A = <id> * (<expr>)
A = <id> * (<id> + <expr>)
A = <id> * (<id> + <id>)
A = A * (B + C)
```

(D) A = B * (C * (A + B))

```
A = <expr>
A = <id> * <expr>
A = <id> * (<expr>)
A = <id> * (<id> * <expr>)
A = <id> * (<id> * (<expr>))
A = <id> * (<id> * (<id> + <expr>))
A = <id> * (<id> * (<id> + <id>))
A = B * (C * (A + B))
```

---

5) Resposta

```
<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <expr> + <term> | <term>
<term> → <term> * <factor> | <factor>
<factor> → (<expr>) | <id>
```

(A) A = (A + B) * C

```
A = <expr>
A = <term>
A = <term> * <factor>
A = <factor> * (<expr>)
A = <id> * (<expr> + <term>)
A = <id> * (<factor> + <factor>)
A = <id> * (<id> + <id>)
A = (A + B) * C
```

(B) A = B + C + A

```
A = <expr>
A = <expr> + <term>
A = <expr> + <factor>
A = <expr> + <expr> + <term>
A = <term> + <term> + <factor>
A = <factor> + <factor> + <id>
A = <id> + <id> + <id>
A = B + C + A
```

(C) A = A * (B + C)

```
A = <expr>
A = <term>
A = <term> * <factor>
A = <term> * (<expr>)
A = <factor> * (<expr> + <term>)
A = <id> * (<term> + <factor>)
A = <id> * (<factor> + <id>)
A = <id> * (<id> + <id>)
A = A * (B + C)
```

(D) A = B * (C * (A + B))

```
A = <expr>
A = <term>
A = <term> * <factor>
A = <term> * (<term>)
A = <term> * (<term> * <factor>)
A = <term> * (<term> * (<expr>))
A = <term> * (<term> * (<expr> + <term>))
A = <factor> * (<factor> * (<term> + <factor>))
A = <id> * (<id> * (<factor> + <id>))
A = <id> * (<id> * (<id> + <id>))
A = B * (C * (A + B))
```

8) Resposta

```
<S> → <A> a <B> b
<A> → <A> b | b
<B> → a <B> | a
```

Palavra	Gerada?
baab	✓ Sim
bbbab	✗ Não
bbaaaaa	✗ Não
bbaab	✓ Sim

---

9) Resposta

```
<program> → begin <stmt_list> end
<stmt_list> → <stmt> ( ; <stmt> )*
<stmt> → <var> = <expression>
<var> → A | B | C
<expression> → <var> ( + <var> | - <var> )*
```

---

10) Resposta

```
<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <term> ( + <term> | * <term> )*
<term> → (<expr>) | <id>
```
