Universidade Federal da Paraíba - UFPB

Rennan Weslley da Silva Costa

Teoria da Computação – Projeto 2

João Pessoa – PB
Abril 2018

As **Árvores de Análise** do Projeto 2 estão disponíveis em melhor resolução no repositório citado ao final deste relatório.

Respostas

1.

Tem dois tipos: int e void.

```
tipo    :       'int'
        |       'void'
        ;
```

Onde **int** é definido pelos números de 0 à 9:

```
INT :   '0'..'9'+
    ;
```

2.

Não, porque o programa é uma função (que é definida por um tipo, um ID, zero ou mais parâmetros entre parênteses e de zero ou mais comandos entre chaves).

```
programa:       funcao+;

funcao  :       tipo ID '(' params? ')' '{' comando* '}';
```

3.

```
int main() {
    void a;
    void b;

    return a + b;
}
```

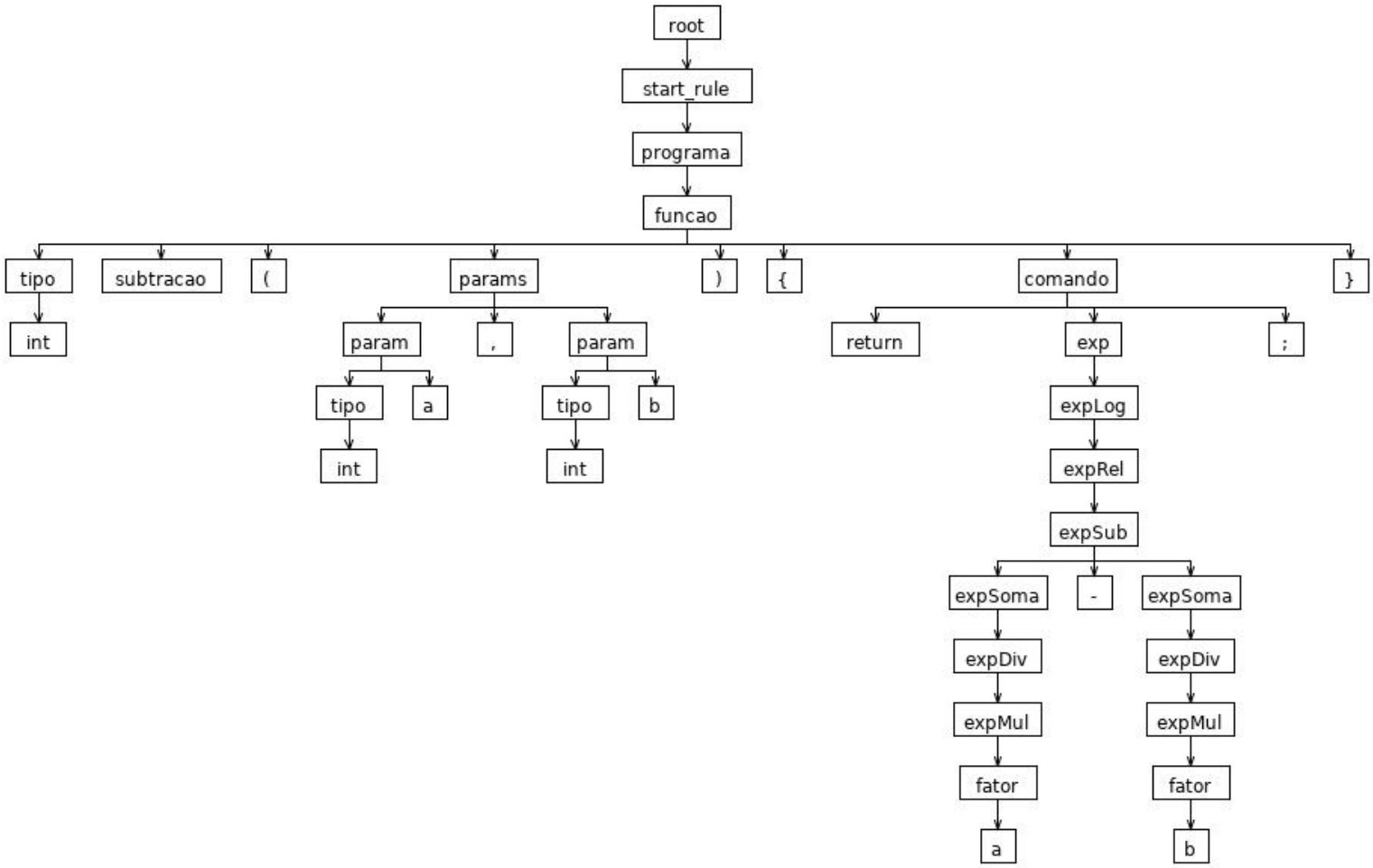
4.

```
expSub  :       expSoma ('-' expSoma)*
        ;
```

Programa:

```
int subtracao(int a, int b) {
   return a - b;
}
```

Árvore de Análise:



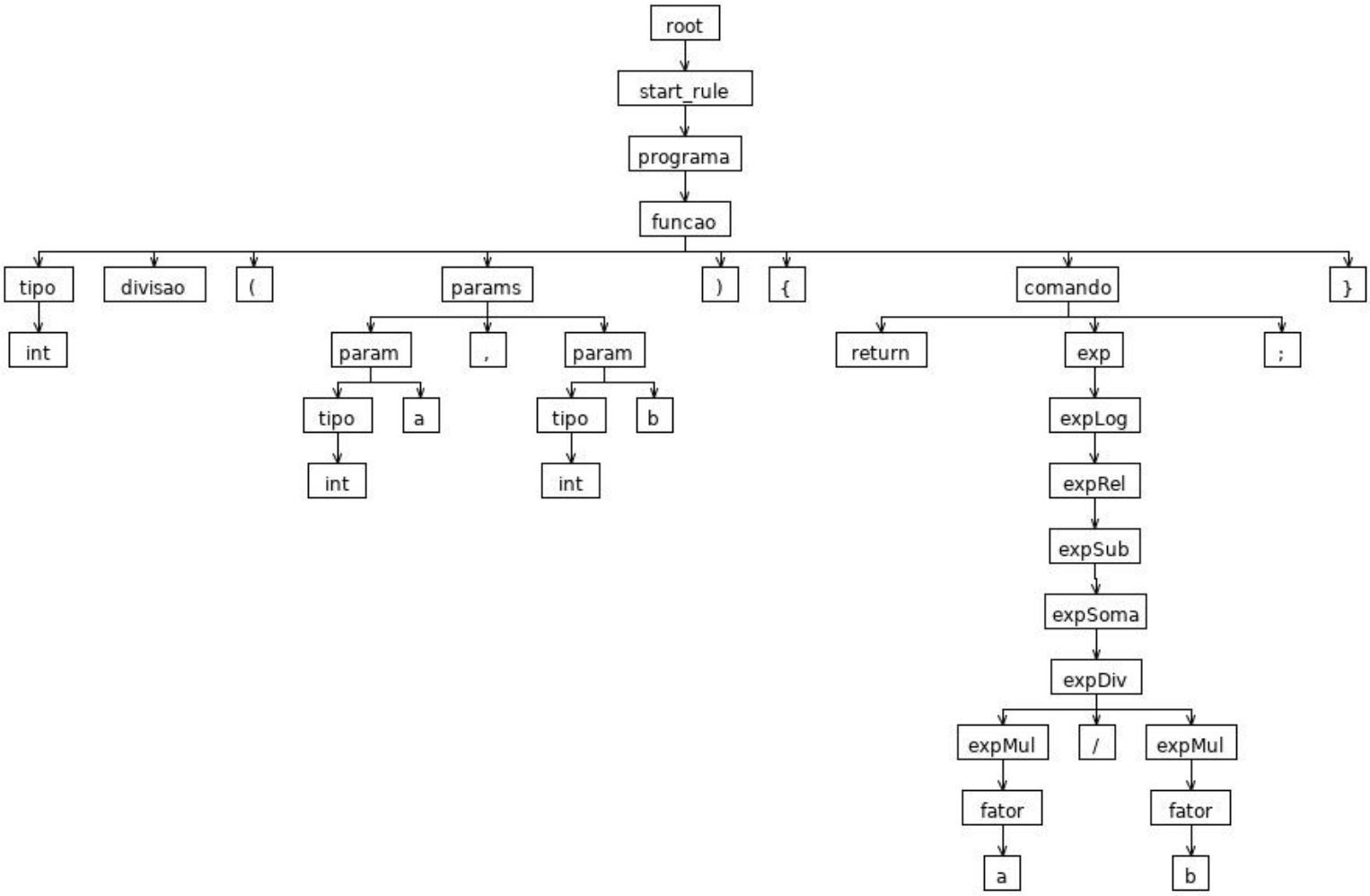
5.

```
expDiv  :       expMul ('/' expMul)*
        ;
```

Programa:

```
int divisao(int a, int b) {
    return a / b;
}
```

Árvore de Análise:



6.

Adicionei a seguinte linha ao não-terminal comando:

```
|          ID '(' args? ');'
```

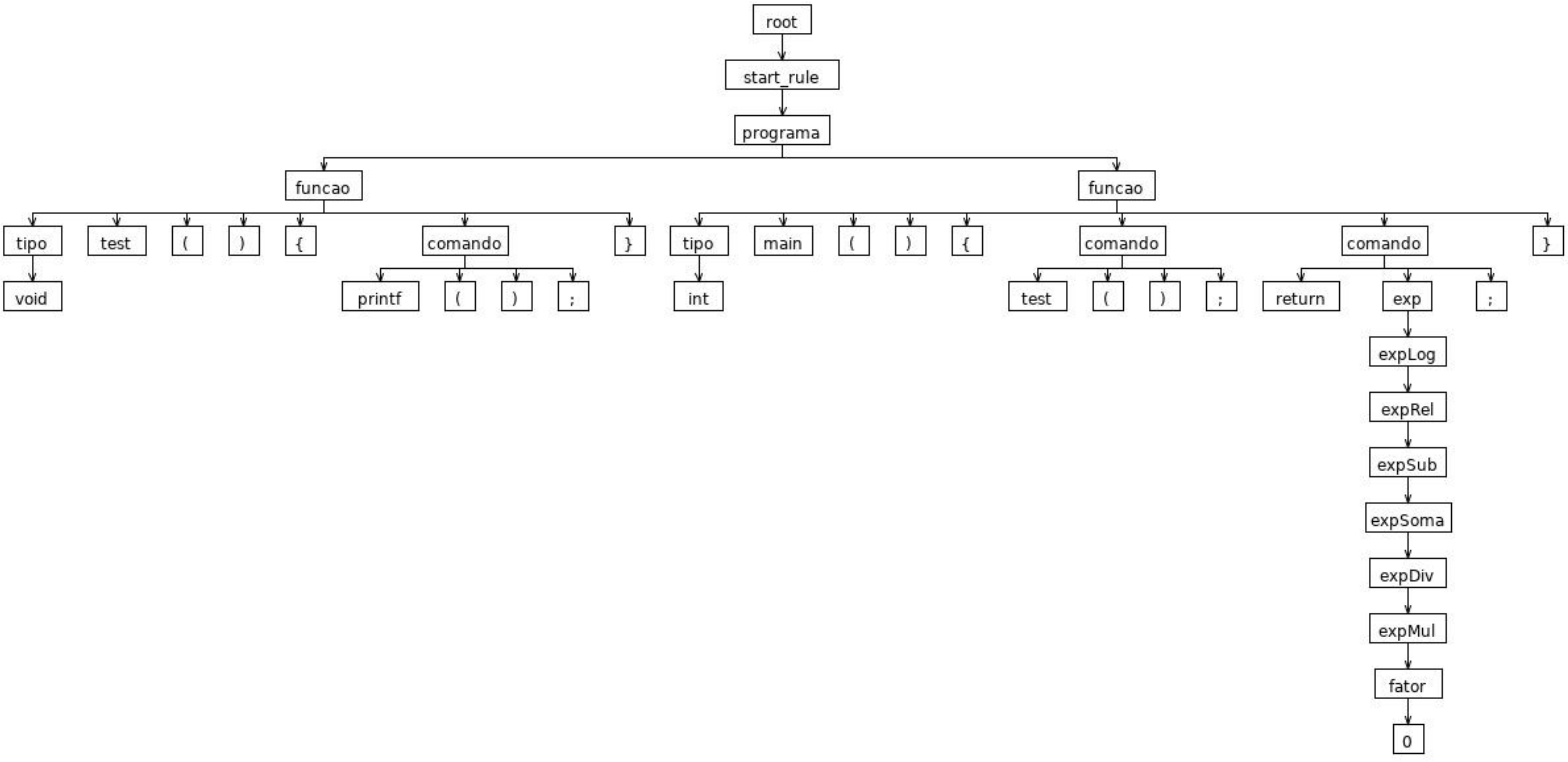
Programa:

```
void test() {
    printf();
}

int main() {
    test();

    return 0;
}
```

Árvore de Análise:



7.

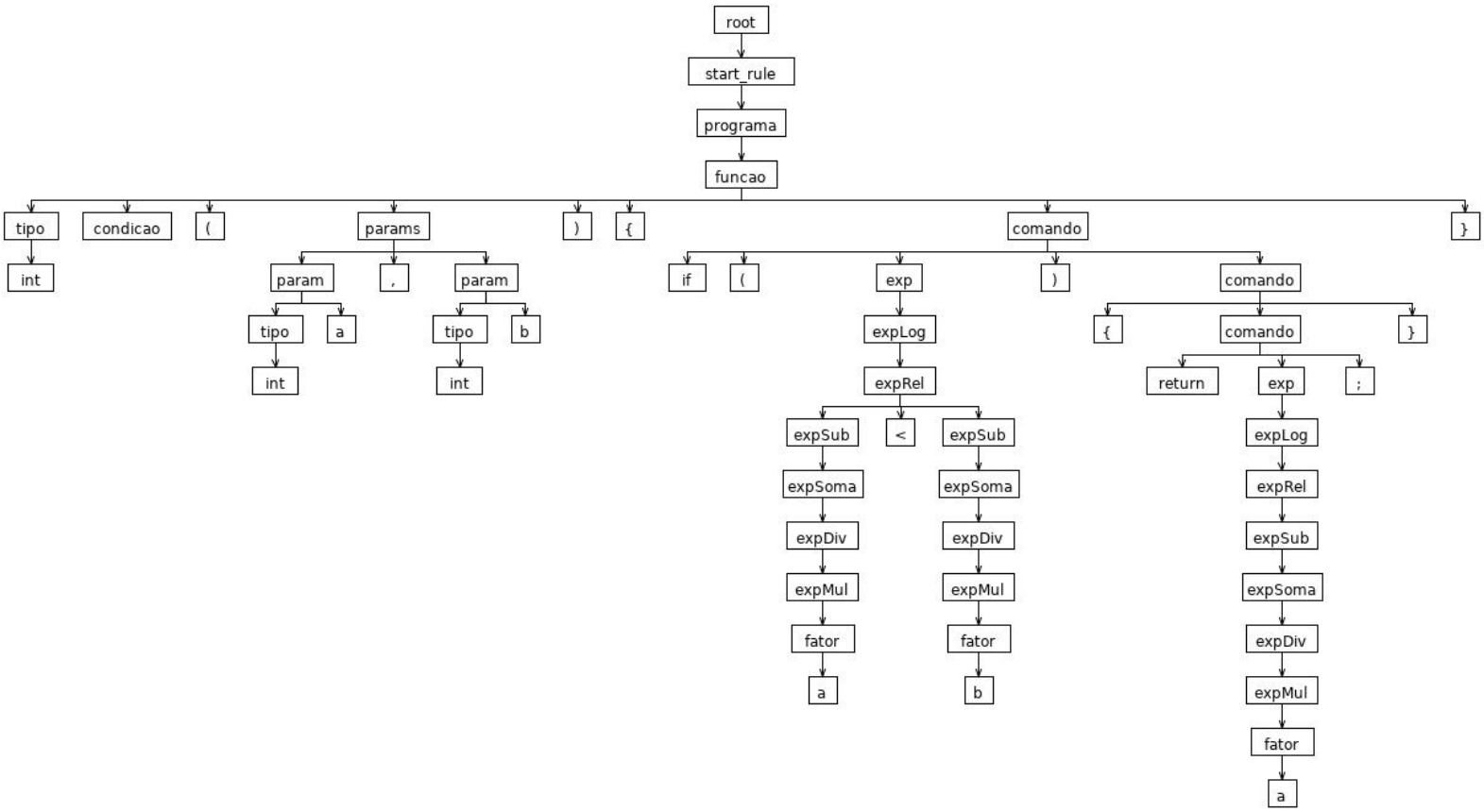
Alterei a linha que correspondia à condição if/else do não-terminal comando para zero ou um ‘else’:

```
| 'if' '(' exp ')' comando ('else' comando)? // condicional
```

Programa:

```
int condicao(int a, int b) {
    if (a < b) {
        return a;
    }
}
```

Árvore de Análise:



8.

...

9.

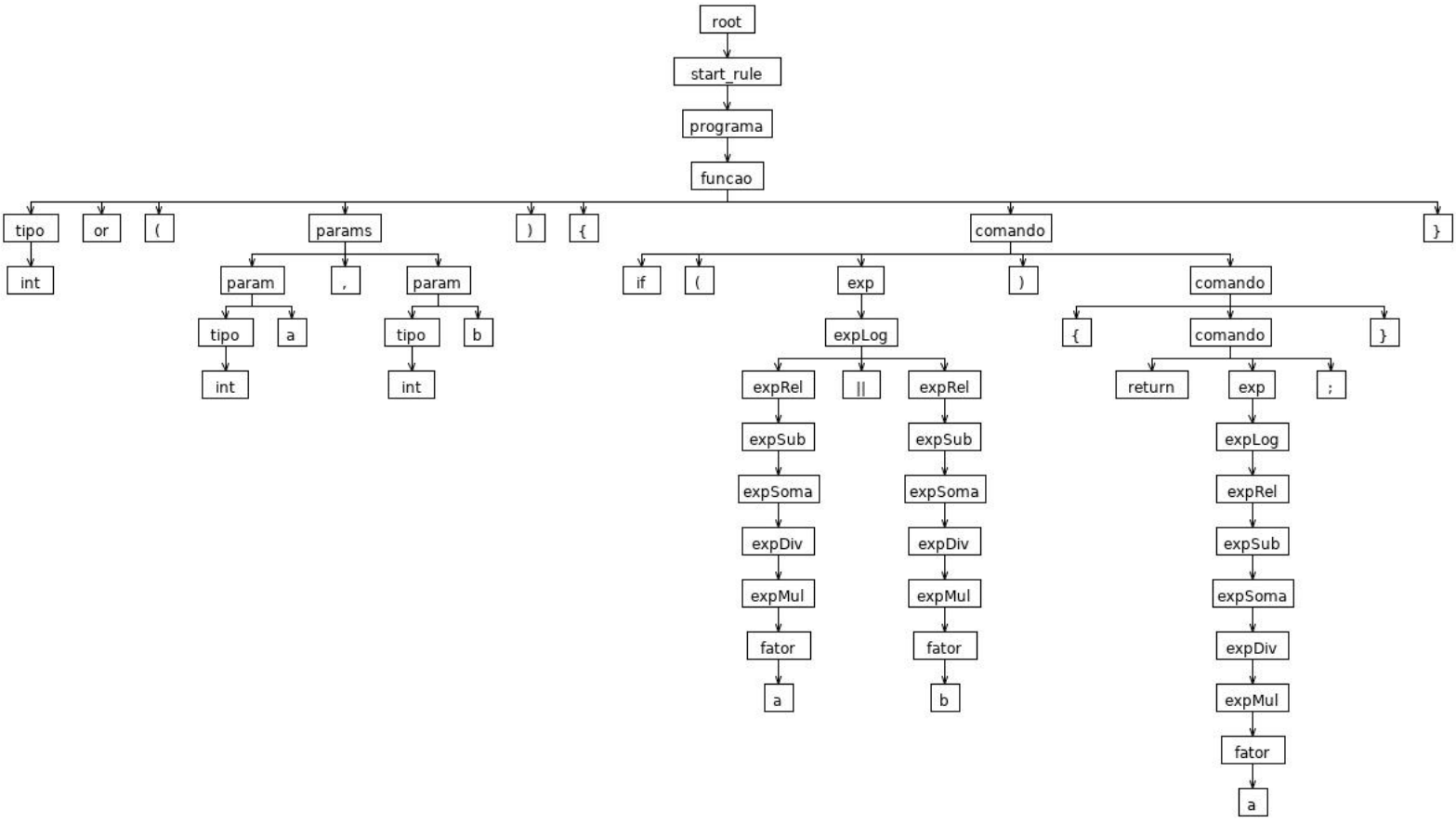
```
exp     :       expLog;

expLog  :       expRel ('||' expRel | '&&' expRel)*
        ;
```

Programa '||':

```
int or(int a, int b) {
    if (a || b) {
        return a;
    }
}
```

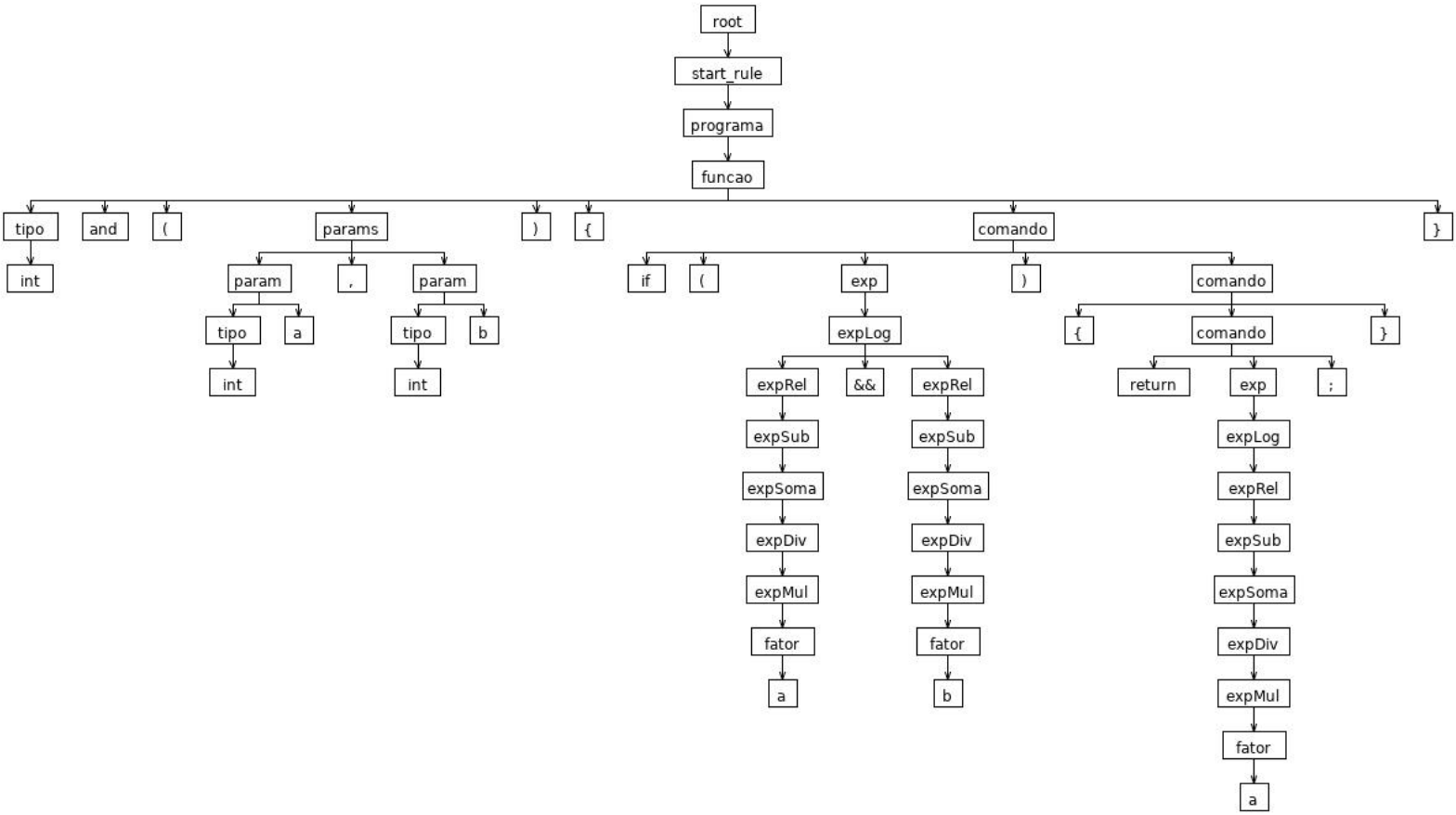
Árvore de Análise '||':



Programa ‘&&’:

```
int and(int a, int b) {
    if (a && b) {
        return a;
    }
}
```

Árvore de Análise ‘&&’:



Código-fonte disponível em: https://github.com/RennanWeslley/Teoria