
Análise Semântica

Análise Semântica

- Verificar se todos os identificadores estão declarados
- Verificar redefinição de identificadores
- Verificar tipos
- Verificar se protótipos correspondem às funções declaradas
- Verificar Expressões matemáticas
- Armazenar informações sobre variáveis
- Verificar se os identificadores estão sendo usados da forma correta
- Verificar o escopo das variáveis
- Construir a AST durante a verificação semântica

Análise Semântica: Escopo

Escopo de um Identificador

O escopo de um identificador é a porção do programa onde este identificador é acessível.

Um mesmo identificador pode se referir a coisas distintas em diferentes partes do programa.

Análise Semântica: Escopo

Escopo de um Identificador

As linguagens de programação podem ter escopo estático ou dinâmico.

Escopo Estático: O escopo do identificador depende apenas do texto do programa fonte. Ex.: Algol, Pascal, C

Escopo Dinâmico: O escopo do identificador depende da execução do programa. Ex.: Lisp, SNOBOL

Análise Semântica: Escopo

Escopo de um Identificador

```
int a;
```

```
int foo(int foo, int a)
{
    return (foo+a);
}
```

O programa está correto?

Análise Semântica: Escopo

Escopo de um Identificador

```
int a;  
char a;
```

```
int foo(int foo, int a)  
{  
    int a;  
    return (foo+a);  
}
```

O programa está correto?

Análise Semântica: Escopo

Escopo de um Identificador

```
int a;  
char a; } identificador a redefinido  
  
int foo(int foo, int a)  
{  
    int a; → identificador a redefinido  
    return (foo+a);  
}
```

Como controlar o escopo dos identificadores e descobrir redefinições dos mesmos?

Tabelas de Símbolos

Tabela de Símbolos: Implementação

Lista dinâmica

| NAME | ATTRIBUTES |
|-------------------|------------|
| | |
| s o r t | |
| a | |
| r e a d a r r a y | |
| i | |
| | |

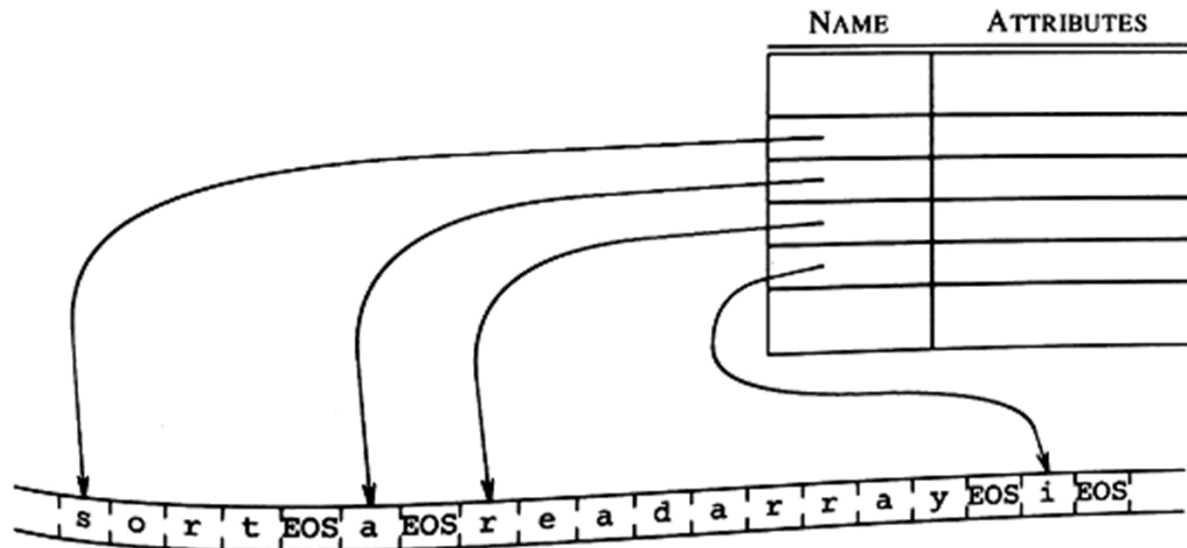


Tabela de Símbolos: Implementação

Tabela Hash

Array of list headers,
indexed by hash value

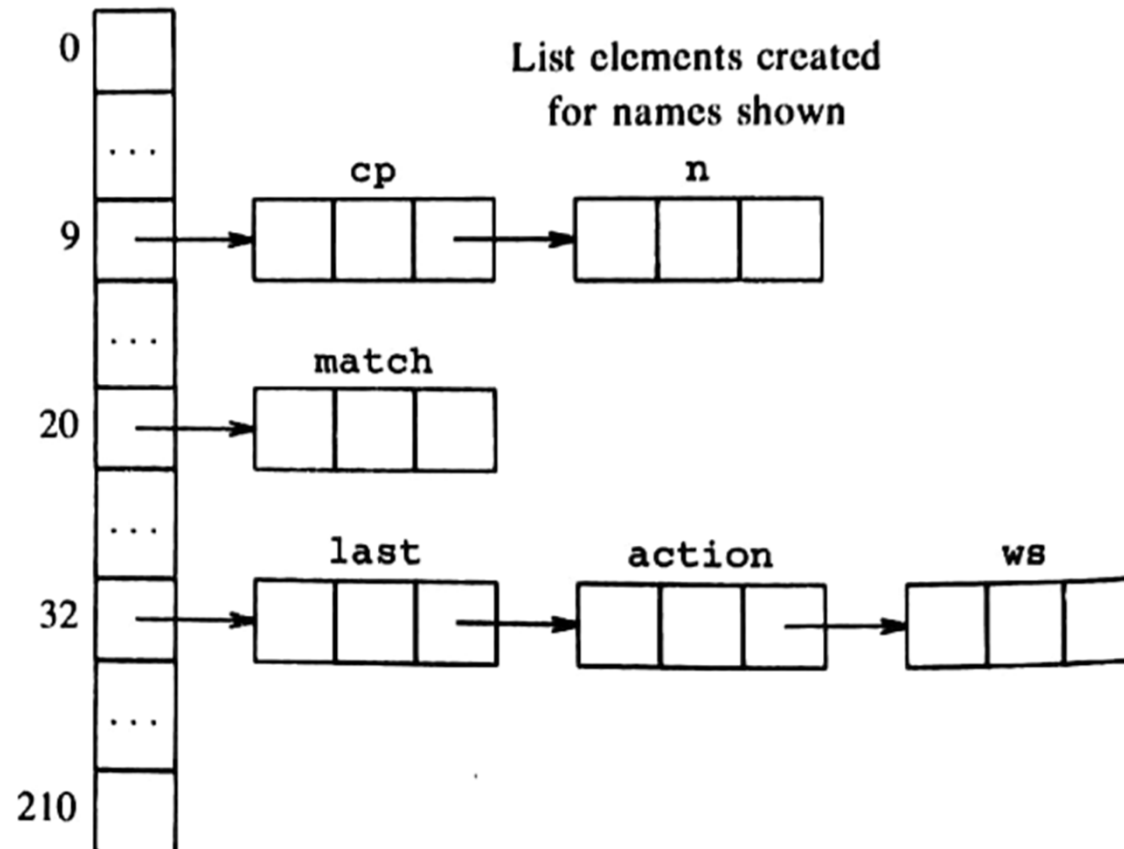


Tabela de Símbolos

Como armazenar/buscar informações?

```
int main()  
{  
    int a, A, chuchu, abobrinha;  
    char abacate, c;  
  
    return 0;  
}
```

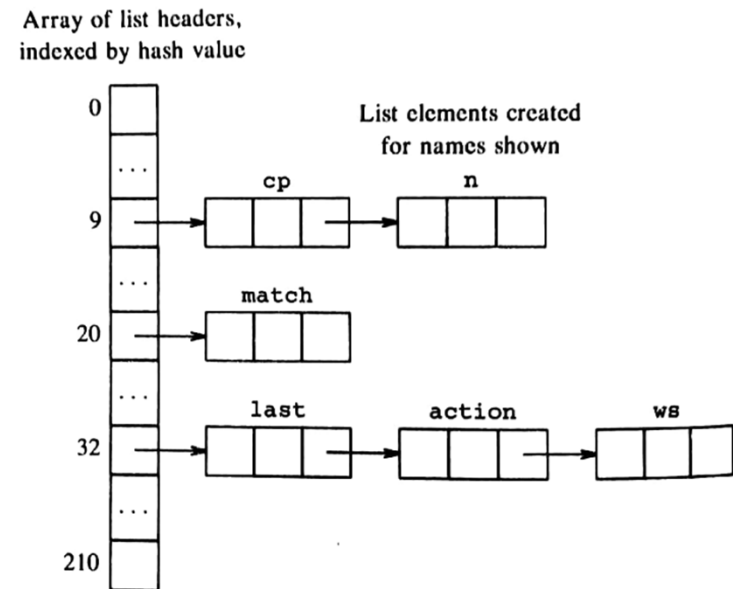


Tabela de Símbolos

chuchu:

Código ASCII dos caracteres:

c = 99

h = 104

u = 117

chuchu = $99+104+117+99+104+117 = 640$

Resto da divisão de 640 por 211 = 7

A posição 7 da tabela hash é utilizada para armazenar o identificador.

Tabela de Símbolos: Para que serve?

Guardar informação para cada identificador que aparece no programa:

```
int a;
int v[10][15];

void func(int a, char* b);

int foo(int foo, int a)
{
    return func(666, "Skynet is Online", v[-1]['a'][77]);
}

int main()
{
    char* c = "AAdeus mundo!" + 1;
    printf("%s\n", c);
    return 0;
}
```

Tabela de Símbolos: O que armazenar?

variáveis: nome, tipo, dimensão, se são locais ou não, se são parâmetros de função.

funções: nome, parâmetros, variáveis locais, tipo de retorno.

protótipos: nome, parâmetros, tipo de retorno.

constantes: nome, valores.

Tabela de Símbolos: Como é usada?

Ao declarar uma variável:

```
int y;
```

- `int` é um tipo válido?
- `y` pode ser declarada nesse ponto?
- Já existe algum outro identificador neste escopo chamado `y`?

Ao declarar uma Função:

```
void function (int k, char* s)
```

- `void` é um tipo válido ?
- Já existe algum outro identificador neste escopo chamado `function`?
- Se existe algum protótipo para `function`, ele corresponde a declaração da função, em relação ao tipo de retorno e aos parâmetros?

Tabela de Símbolos: Como é usada?

Em constantes:

```
#define a 1+5+c;
```

- Já existe algum outro identificador neste escopo chamado **a**?
- O identificador **c** já está declarado e é uma constante?

Em comandos:

```
x[i+9] = y(10, "teste");
```

- O identificador **x** está declarado?
- O identificador **x** é um array? Se sim, ele possui só uma dimensão?
- O identificador **i** está declarado e é do tipo **int**?
- O identificador **y** está declarado?
- O identificador **y** é uma função? Se sim, possui exatamente 2 parâmetros?
- O parâmetros passados para **y** tem o tipo correto?

Criação da AST

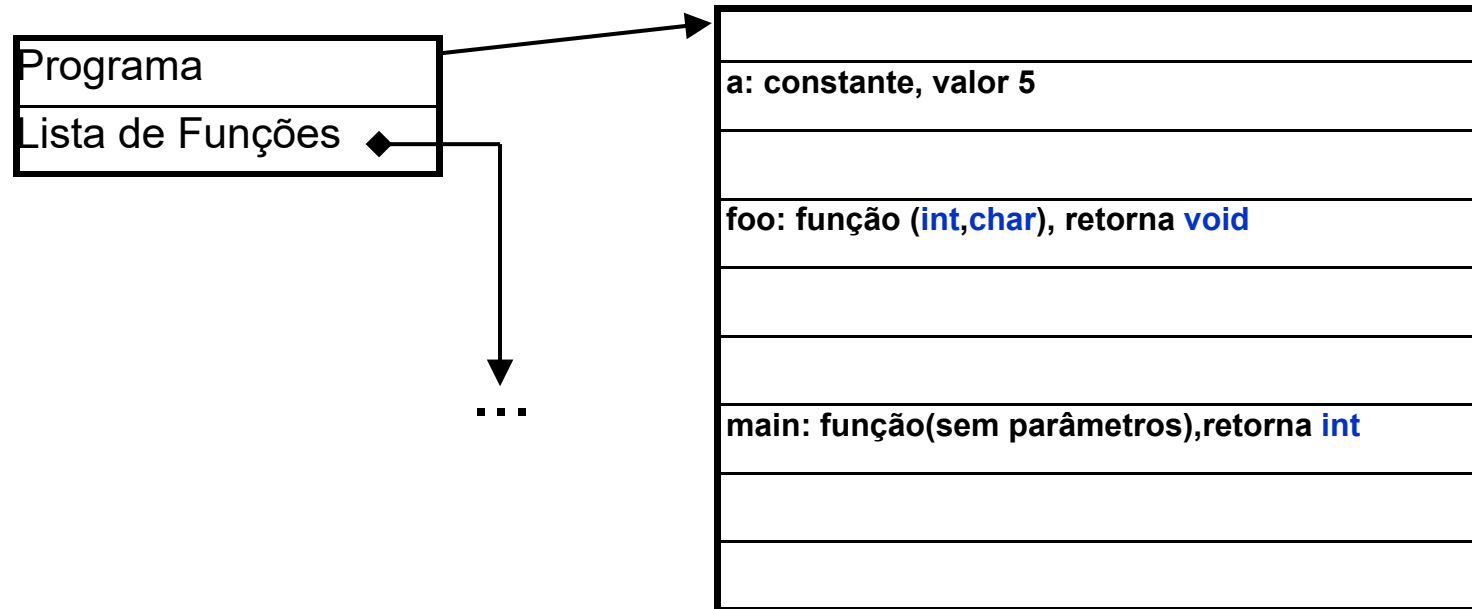
Análise Semântica: Criação da AST

```
#define a 5
```

```
void foo(int a, char b)
{
}
```

```
int main()
{
}
```

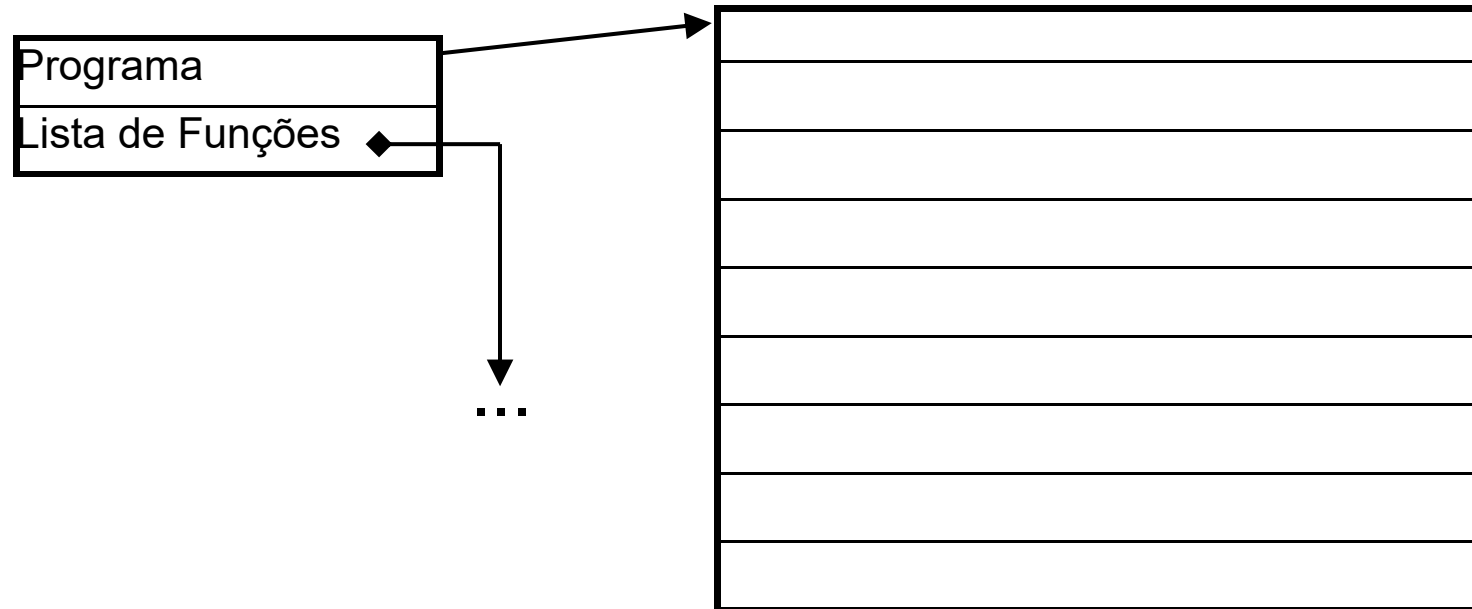
Tabela de Símbolos Global



Análise Semântica: Criação da AST

```
int a;  
char a;  
  
int foo(int foo, int a)  
{  
    int a;  
    return (foo+a);  
}
```

Tabela de Símbolos Global



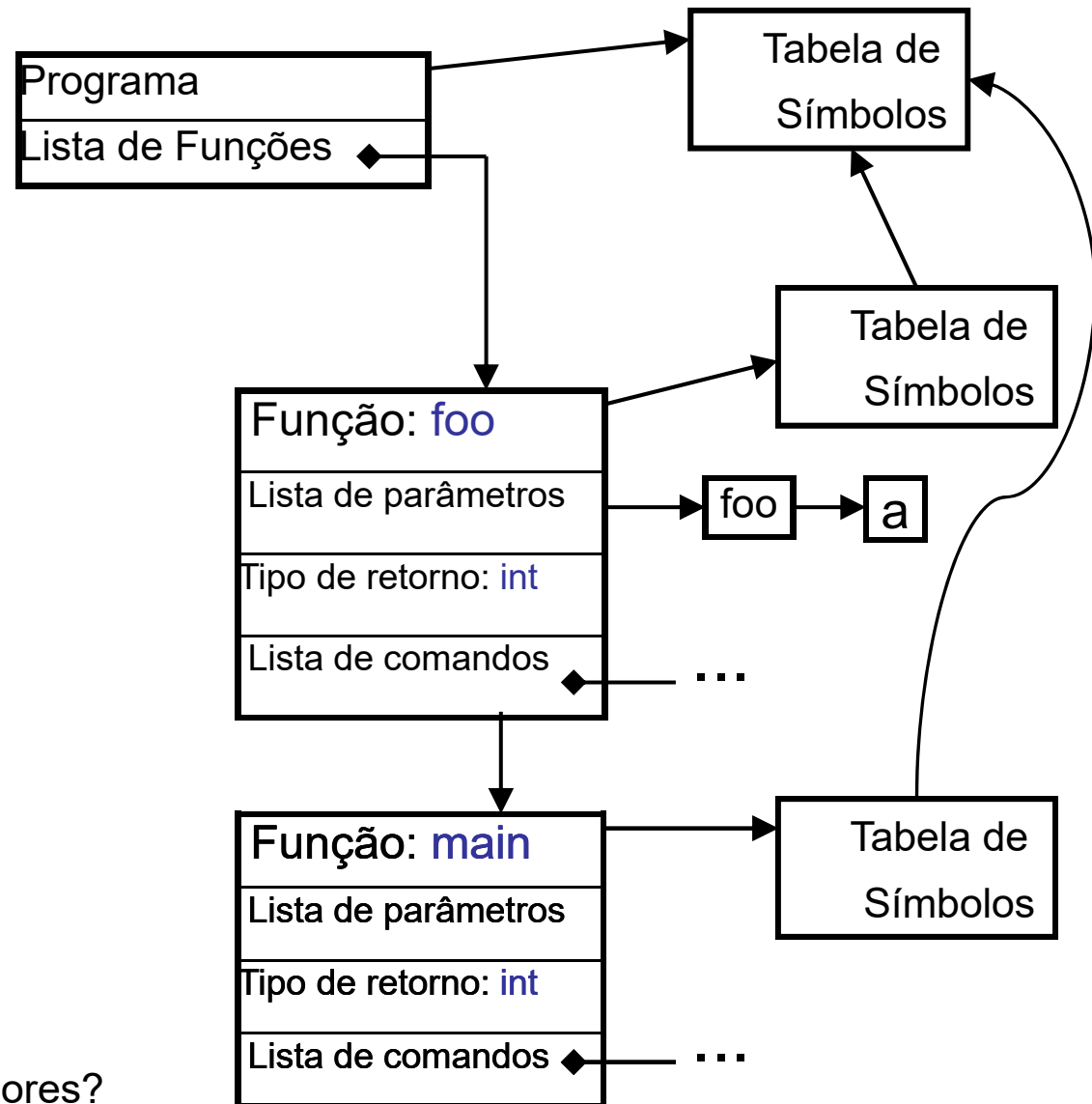
Como fica a inserção dos identificadores?

Análise Semântica: Criação da AST

```
#define c1 5
#define c2 c1+10
int a,b,c[5][5];

int foo(int foo, int a)
{
    int a;
    return (foo+a);
}

int main()
{
}
```



Como fica a inserção dos identificadores?

Análise Semântica: Criação da AST

Cada elemento da linguagem fonte, vai possuir uma estrutura na árvore para o representar.

Tal estrutura pode ser, por exemplo, uma `struct` ou uma classe.

Análise Semântica: Criação da AST

Comando IF

Análise Semântica: Criação da AST

Comando IF

- Expressão matemática
- Lista de comandos para THEN
- Lista de comandos para ELSE

| |
|--------|
| IF |
| exp → |
| then → |
| else → |

Análise Semântica: Criação da AST

Comando WHILE

Análise Semântica: Criação da AST

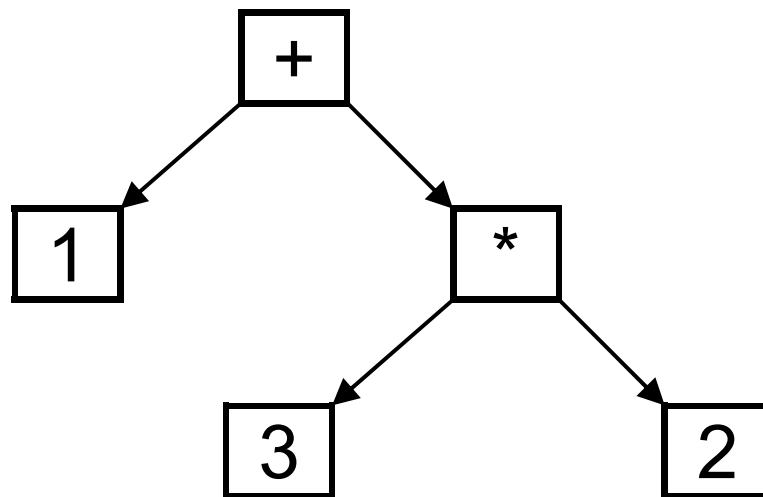
Comando WHILE

- Expressão matemática
- Lista de comandos

| |
|------------|
| WHILE |
| exp → |
| comandos → |

Análise Semântica: Criação da AST

Expressão Matemática: $1+3*2$



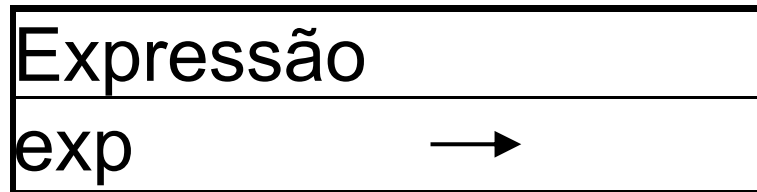
Análise Semântica: Criação da AST

Atribuição: $a = 1 + 3 * 2;$

Análise Semântica: Criação da AST

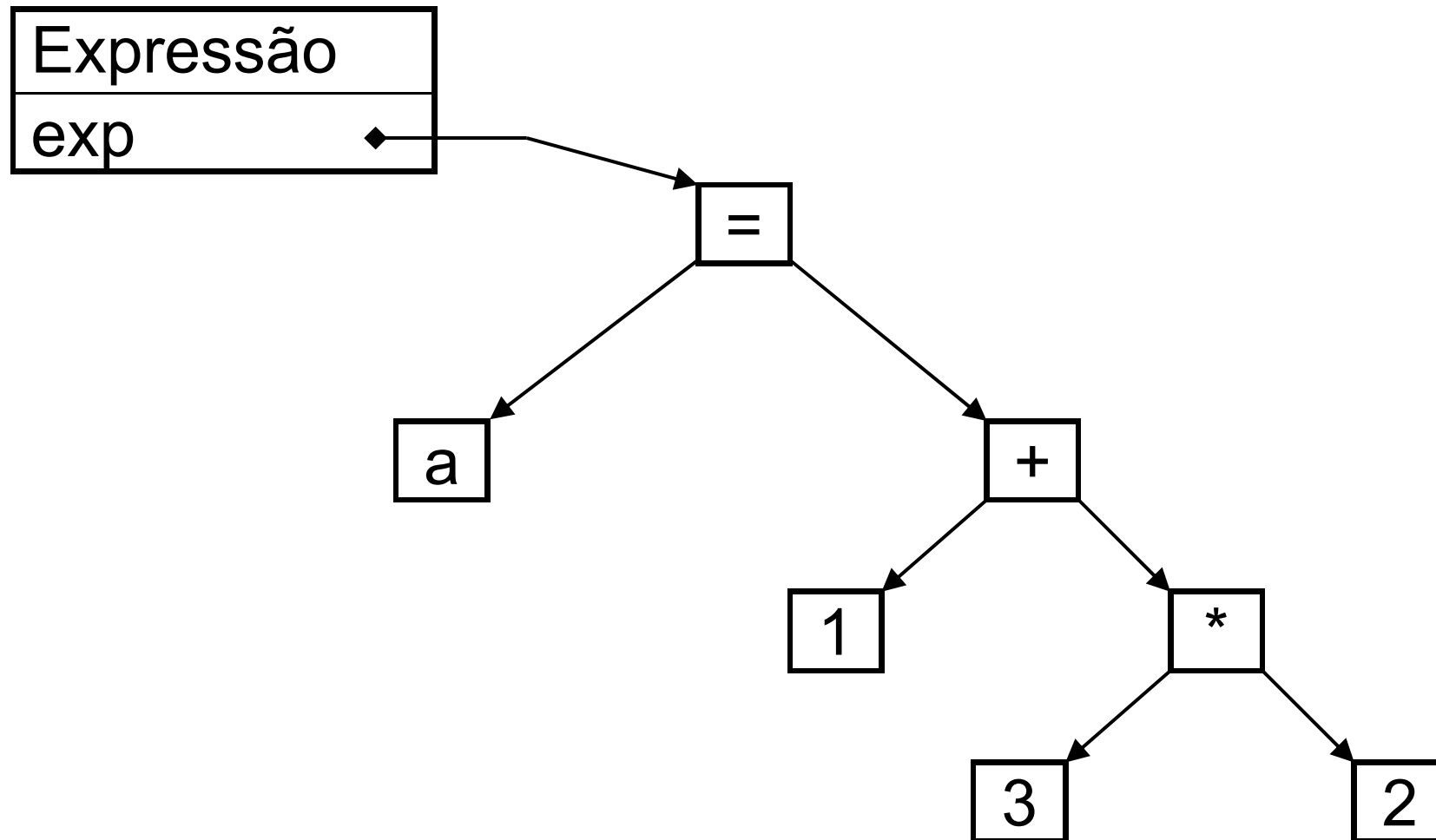
Atribuição: $a = 1 + 3 * 2;$

- Expressão



Análise Semântica: Criação da AST

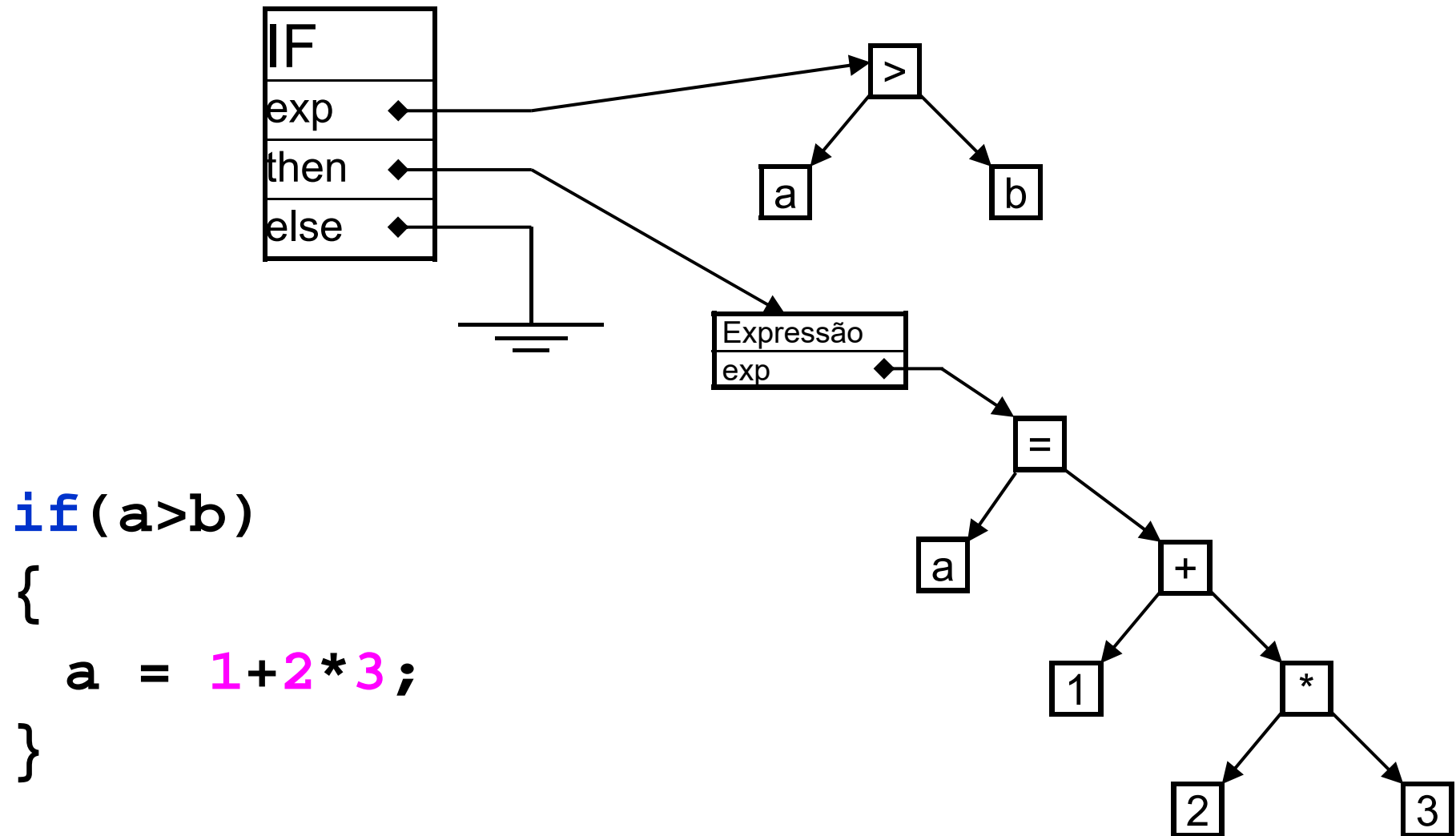
Atribuição: $a = 1 + 3 * 2;$



Análise Semântica: Criação da AST

```
if (a > b)
{
    a = 1 + 2 * 3;
}
```

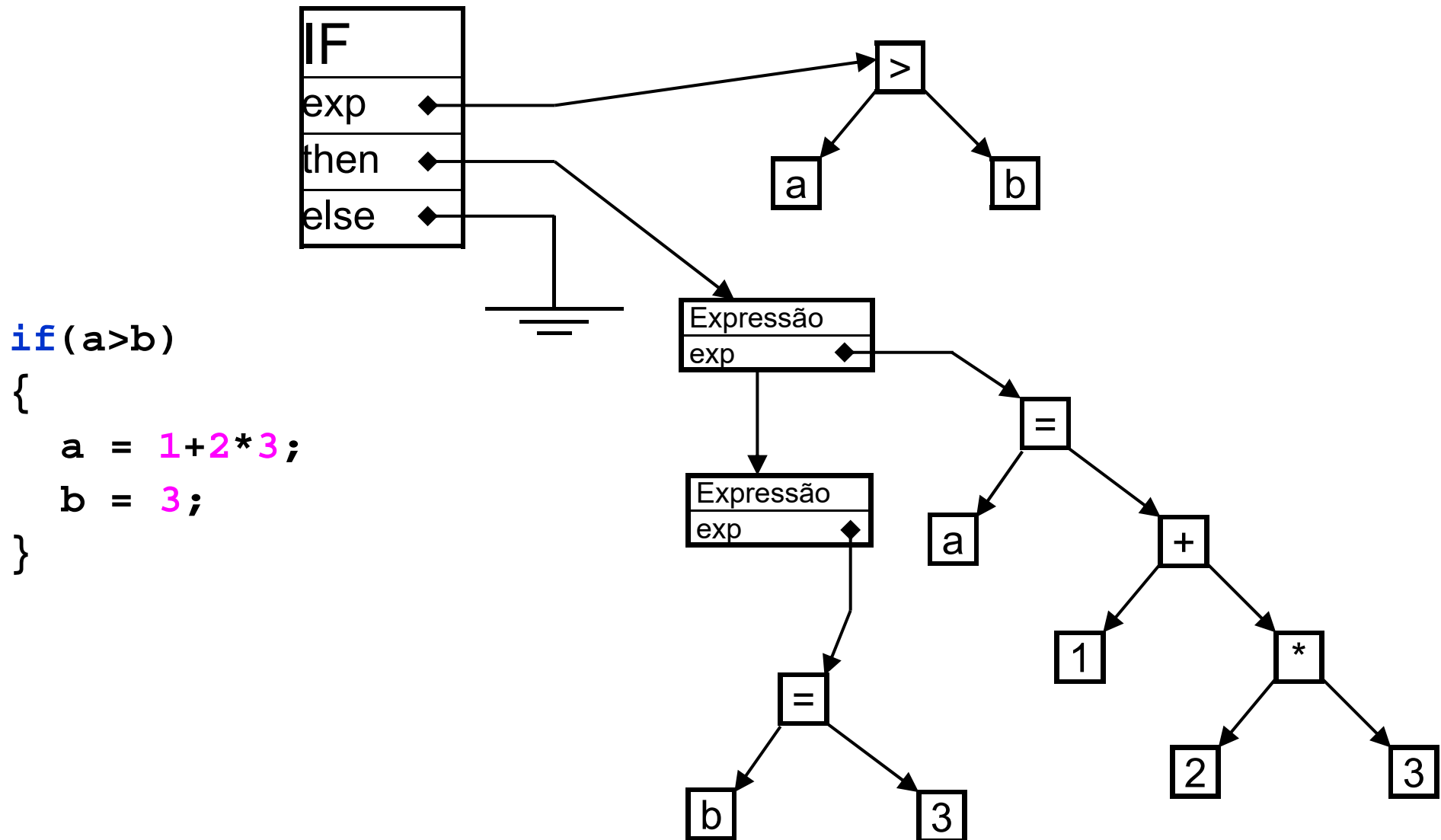
Análise Semântica: Criação da AST



Análise Semântica: Criação da AST

```
if(a>b)
{
    a = 1+2*3;
    b = 3;
}
```

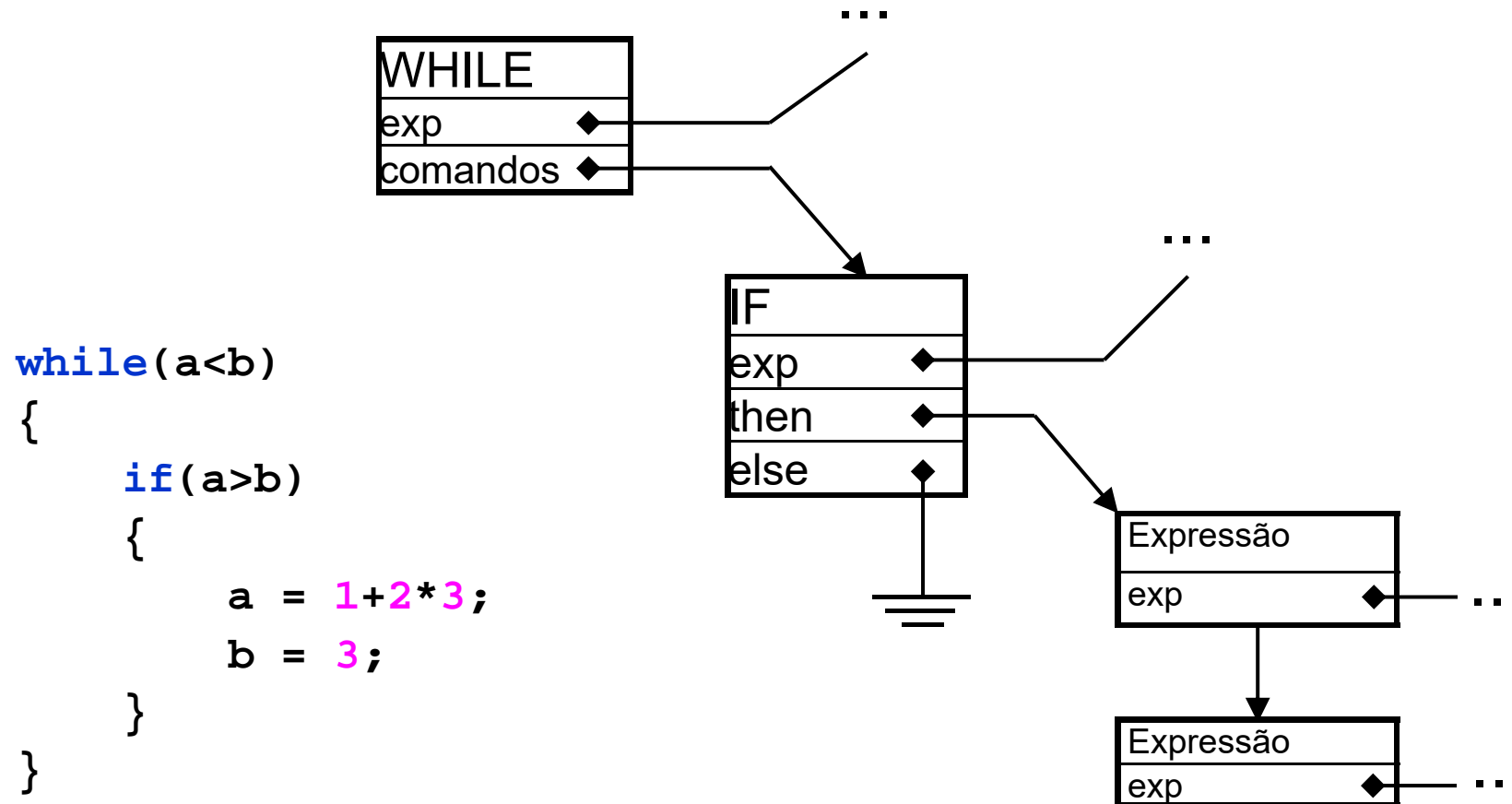

Análise Semântica: Criação da AST



Análise Semântica: Criação da AST

```
while(a<b)
{
    if(a>b)
    {
        a = 1+2*3;
        b = 3;
    }
}
```

Análise Semântica: Criação da AST



Análise Semântica: Criação da AST

Como seria a AST para:

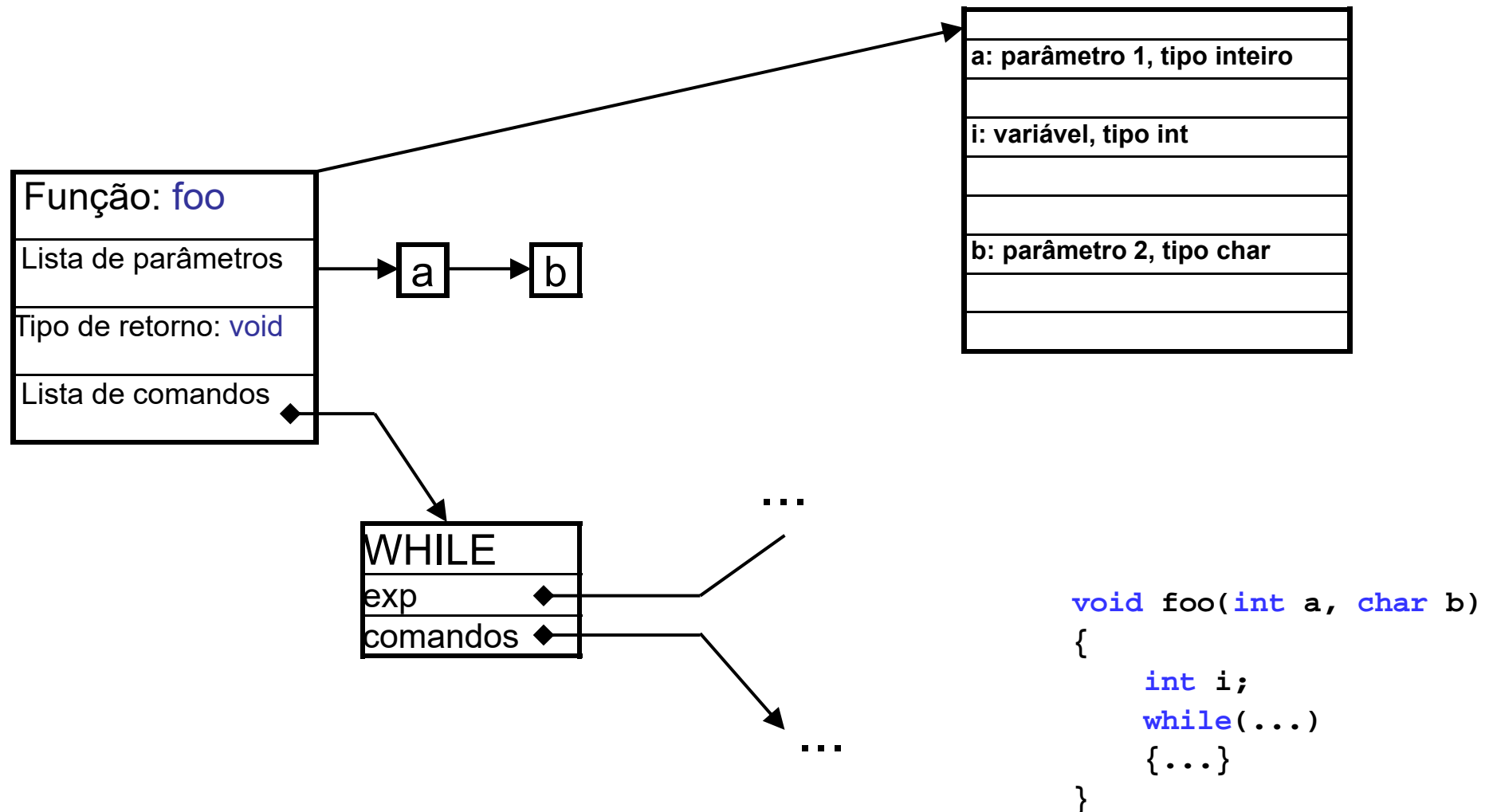
```
if(a>b)
{
    a = 1+2*3;
    b = 3;
}
else
{
    if(1==b)
    {
        c = e+1;
    }
}
```

Análise Semântica: Criação da AST

```
void foo(int a, char b)
{
    int i;
    while(...)
    {...}
}
```

Análise Semântica: Criação da AST

Tabela de Símbolos



Análise Semântica: Criação da AST

Como seria a AST para:

```
#define MAX 10
int i;
void bar(int i)
{
    if(bar(3))
    {
        i = a+MAX;
    }
}

int main()
{
    a = bar('a', 3);
    bar = 4;
    return MAX;
}
```