

ANÁLISE SEMÂNTICA

- O que é?
 - Análise do “contexto”
 - Semelhante à análise do “significado” de um texto
 - Validação final de um programa, relativamente à especificação da linguagem
- Em que consiste?
 - Resolução das referências (nomes)
 - Completar e validar a informação de tipos
- Como se processa?
 - Sobre a APT
 - Análise de nomes
 - Análise de tipos
 - Ambas em simultâneo

RESOLUÇÃO DE NOMES

- Um nome refere alguma coisa em particular
- Não confundir o nome com aquilo que designa
- Analisar um nome é:
 - Desreferenciar o nome, ou seja, ligar (ou substituir) o nome àquilo que este representa

ANÁLISE DE TIPOS

- Determinar o domínio para uma expressão ou variável
- Com isto, conseguimos saber quais as operações admissíveis sobre a expressão ou variável
- Consiste em:
 - Preenchimento da informação de tipo para cada nó da APT
 - Verificação da coerência de tipos
 - P.ex. uma soma deve ser feita entre expressões de tipos "compatíveis"
 - P.ex. a condição de um *if-then-else* deve ser booleana (ou "compatível")
 - P.ex. um *return* deve respeitar o tipo declarado da função onde ocorre

PROCESSO

- Travessia da APT
 - De uma só vez
 - Processo recursivo
 - Cada nó tem uma estrutura esperada, que conduz a análise

SYMBOL TABLE

○ Dicionário

- Associa nomes a estruturas de dados
- Pares (chave, valor)
- Implementação usando estruturas apropriadas (árvores, hash tables, ...)
- Diversos usos
 - Declaração de variáveis (globais, locais, argumentos)
 - Definição de novos tipos
 - Nomes de atributos/campos/métodos (para classes, structs)
 - Outras propriedades menos evidentes

Tamanho na memória (storage size)

Posição na stack frame

SYMBOL TABLE - IMPLEMENTAÇÃO

○ Tipo “abstracto”

- Associação entre “chaves” e “valores”
 - A cada “chave” corresponde um “valor” único
 - Considerando a “ocultação” de variáveis de um âmbito superior
- ```
int a = 20; /* variável global "a" */
void func(char a[]) /* argumento oculta a variável
global */ . . .
```

## SYMBOL TABLE - OPERAÇÕES

- *insert* - criação de um novo par (chave, valor)
- *lookup* - dada uma chave, obter o valor associado
- *new environment* - criação de um novo “contexto” de ocultação
- *drop environment* - reposição do contexto anterior (quando da saída da função, por exemplo)

## ANÁLISE DE NOMES

- Objectivo
  - Substituir ocorrências de nomes por referências directas às entidades que estes referem
- São encontrados alguns erros
  - Uso de nomes sem prévia declaração
  - Definições múltiplas para o mesmo nome
  - Não há ainda conhecimento “fino” (completo) das consequências de uma definição (p.ex. tipos compostos)
- Processo
  - Cada referência é “desreferenciada”, ou seja, substituída pelo seu “objecto” referido

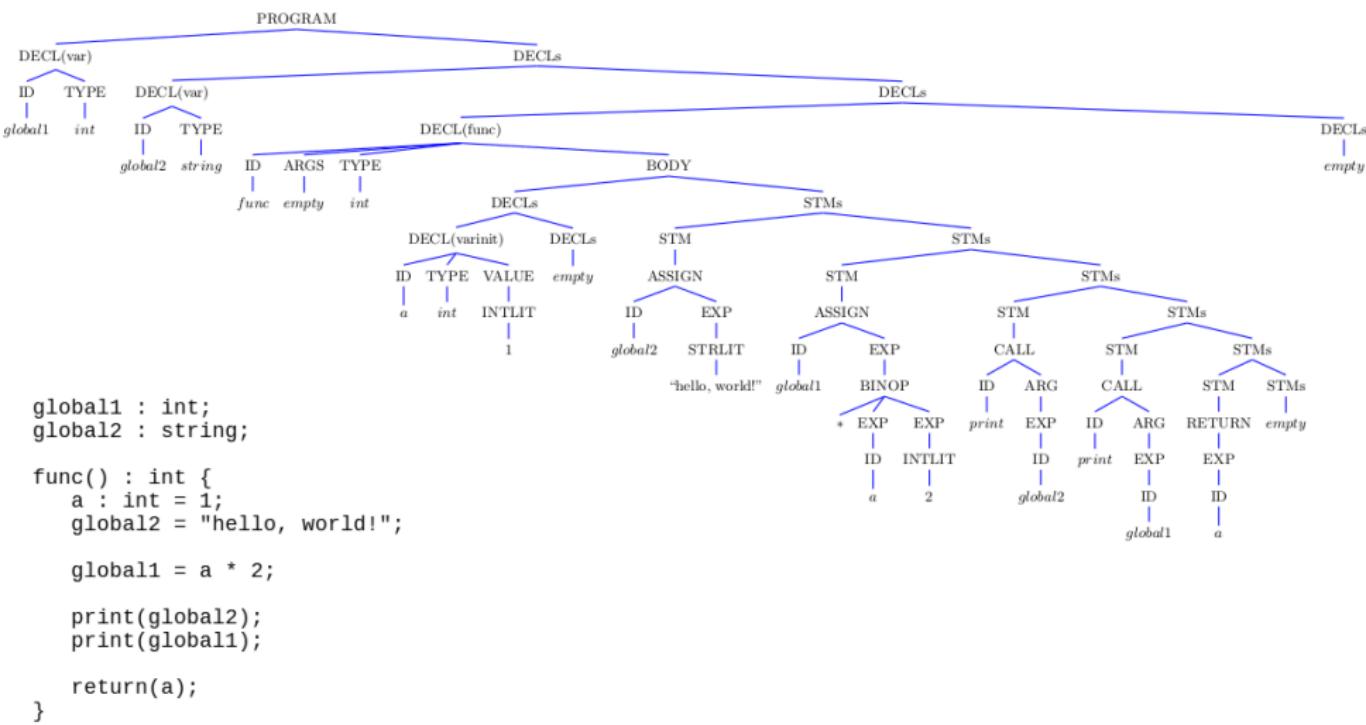
## ANÁLISE DE NOMES - PROCESSO

- Tipo de ocorrência
  - Definição  
Acrescenta-se à Symbol Table (*insert*)
  - Uso  
Substitui-se a referência pela definição presente na Symbol Table (*lookup*)
- Linguagens “mal-comportadas”
  - Variáveis podem não ser declaradas explícitamente

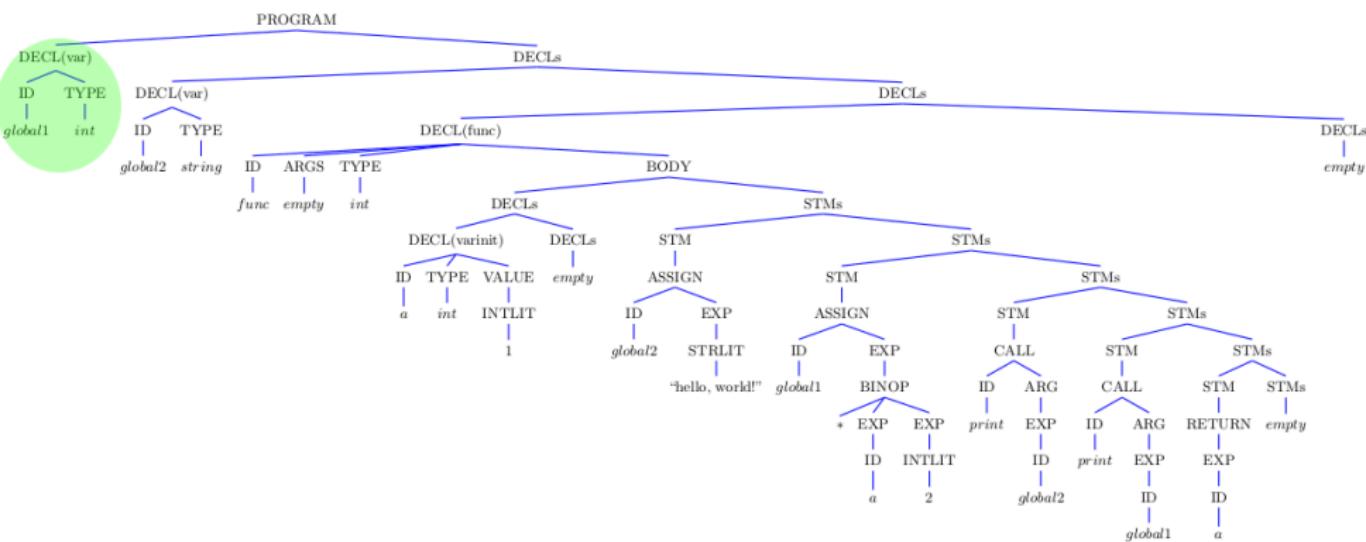
## ANÁLISE DE TIPOS

- Processamento, na APT, dos nós que necessitam de informação de tipo
- Preenchimento da informação de tipo
  - No caso desta estar omissa
  - Para linguagens com inferência de tipos (Python, PHP, ...)
- Verificação da adequação dos tipos encontrados a cada situação de uso
  - Conversão de tipos “compatíveis” para o tipo mais “amplo”
  - Detecção de erros de tipo (tipos “incompatíveis”)

# SYMBOL TABLE - EXEMPLO



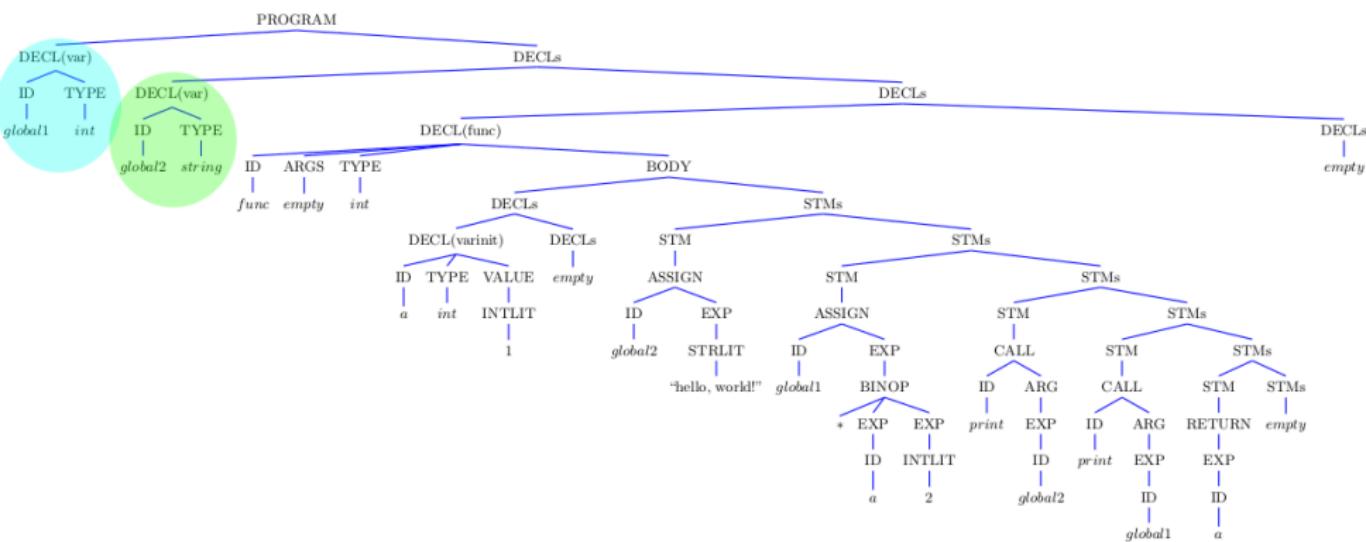
# SYMBOL TABLE - EXEMPLO



**ID            TYPE**

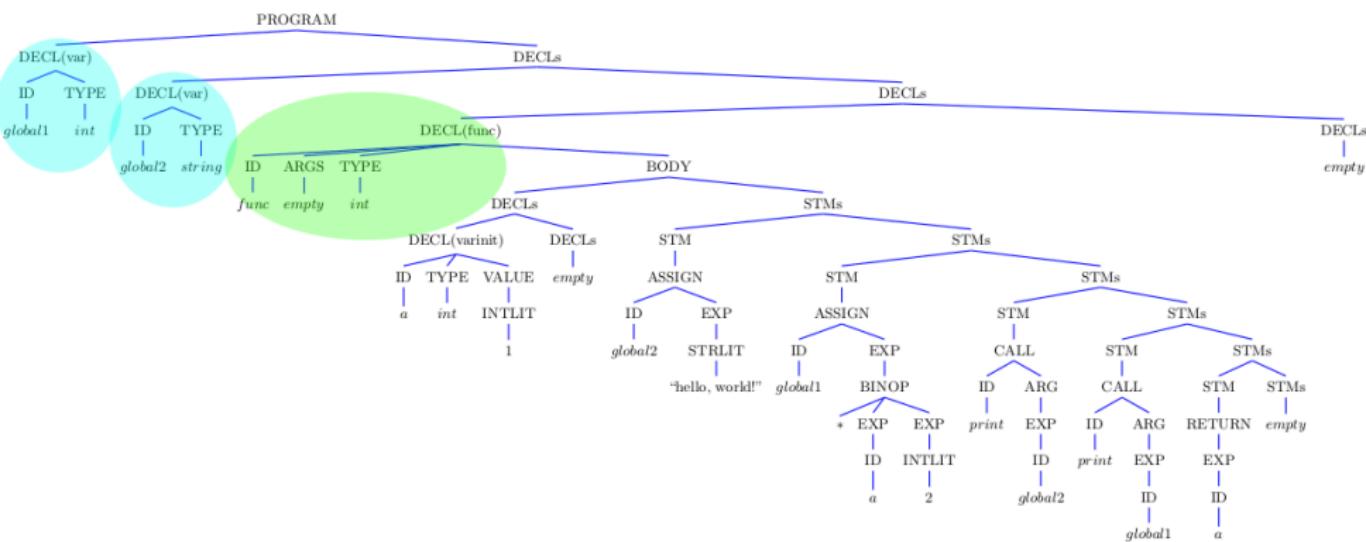
|         |     |                      |
|---------|-----|----------------------|
| global1 | int | insert(global1, int) |
|---------|-----|----------------------|

## SYMBOL TABLE - EXEMPLO



| ID      | TYPE   |                         |
|---------|--------|-------------------------|
| global1 | int    | insert(global2, string) |
| global2 | string |                         |

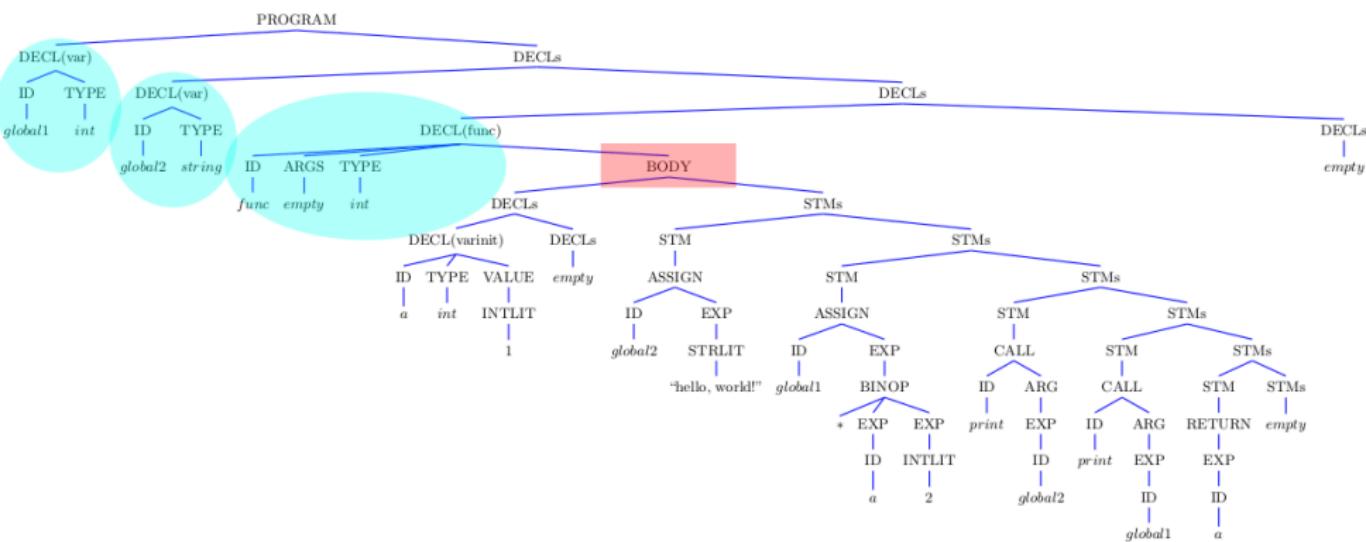
# SYMBOL TABLE - EXEMPLO



## ID      TYPE

|         |        |      |                           |
|---------|--------|------|---------------------------|
| global1 | int    |      | insert(func, int, [void]) |
| global2 | string |      |                           |
| func    | int    | void |                           |

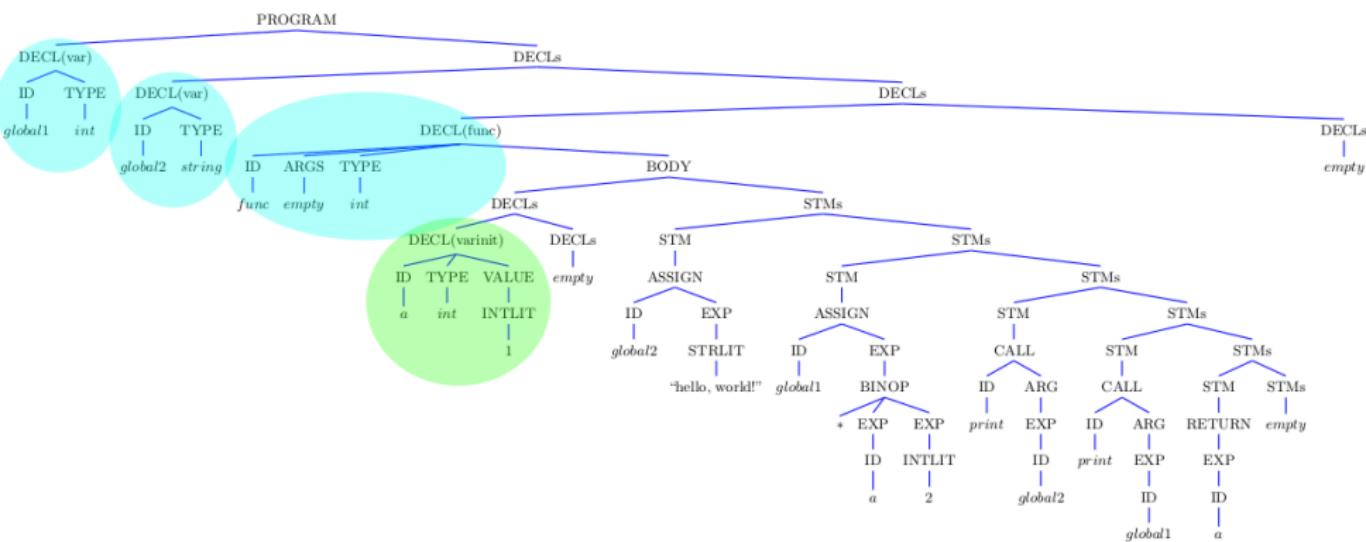
# SYMBOL TABLE - EXEMPLO



| ID      | TYPE   | ARGS |
|---------|--------|------|
| global1 | int    |      |
| global2 | string |      |
| func    | int    | void |

`new_scope()`

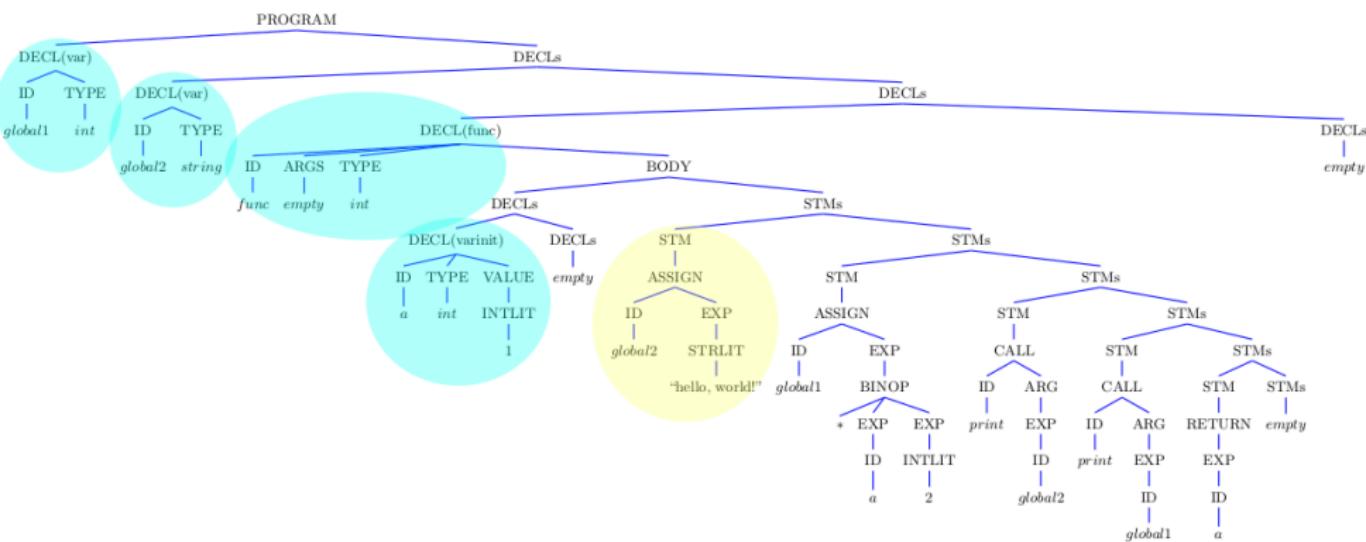
# SYMBOL TABLE - EXEMPLO



| ID | TYPE | ARGS |
|----|------|------|
|----|------|------|

|         |        |      |                |
|---------|--------|------|----------------|
| global1 | int    |      | insert(a, int) |
| global2 | string |      |                |
| func    | int    | void |                |
| <hr/>   |        |      |                |
| a       | int    |      |                |

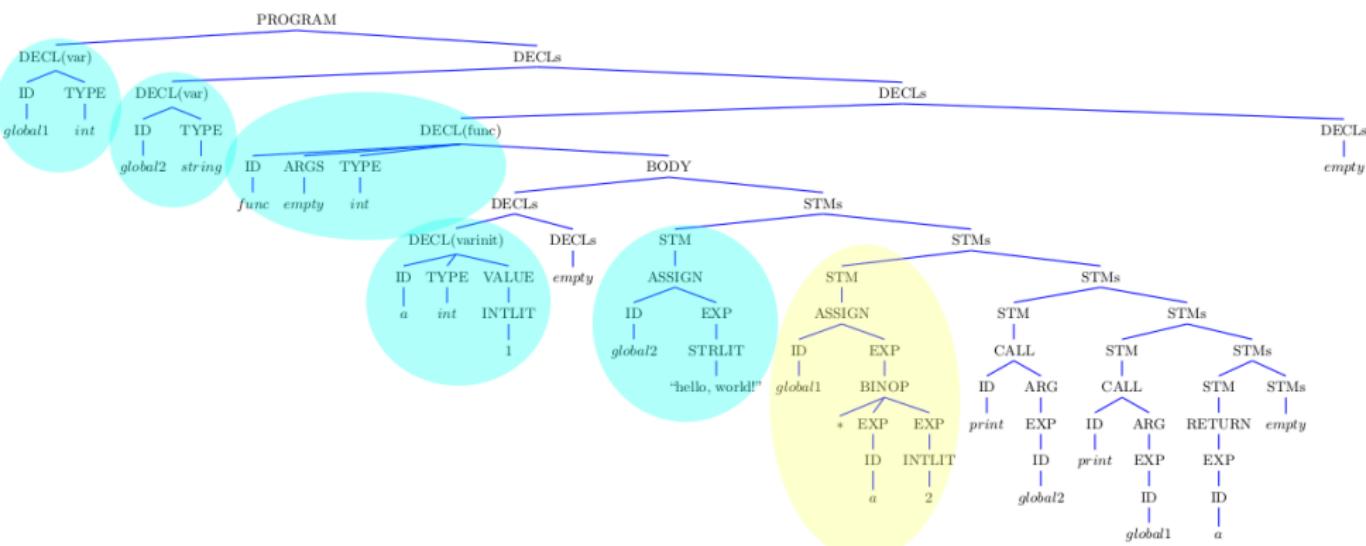
## SYMBOL TABLE - EXEMPLO



| ID      | TYPE   | ARGS |
|---------|--------|------|
| global1 | int    |      |
| global2 | string |      |
| func    | int    | void |
| a       | int    |      |

lookup(global2)

# SYMBOL TABLE - EXEMPLO



| ID | TYPE | ARGS |
|----|------|------|
|----|------|------|

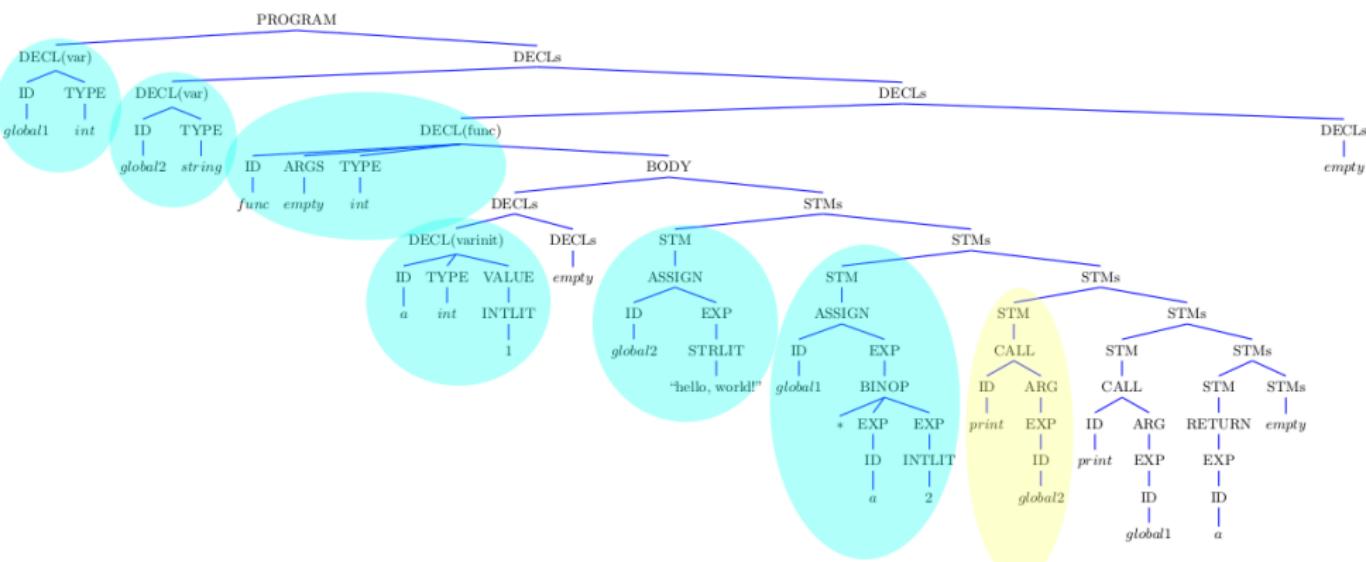
|         |        |  |
|---------|--------|--|
| global1 | int    |  |
| global2 | string |  |

|      |     |      |
|------|-----|------|
| func | int | void |
|------|-----|------|

|   |     |
|---|-----|
| a | int |
|---|-----|

lookup(global1)  
lookup(a)

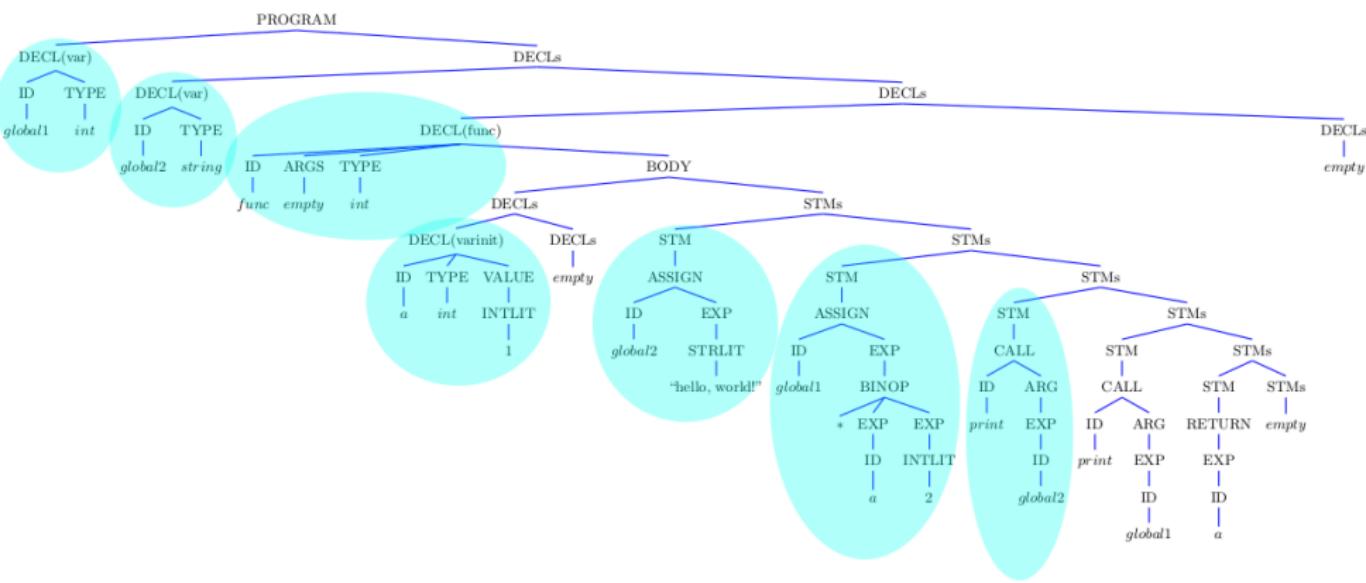
## SYMBOL TABLE - EXEMPLO



| ID      | TYPE   | ARGS |
|---------|--------|------|
| global1 | int    |      |
| global2 | string |      |
| func    | int    | void |
| a       | int    |      |

lookup(print)  
lookup(global2)

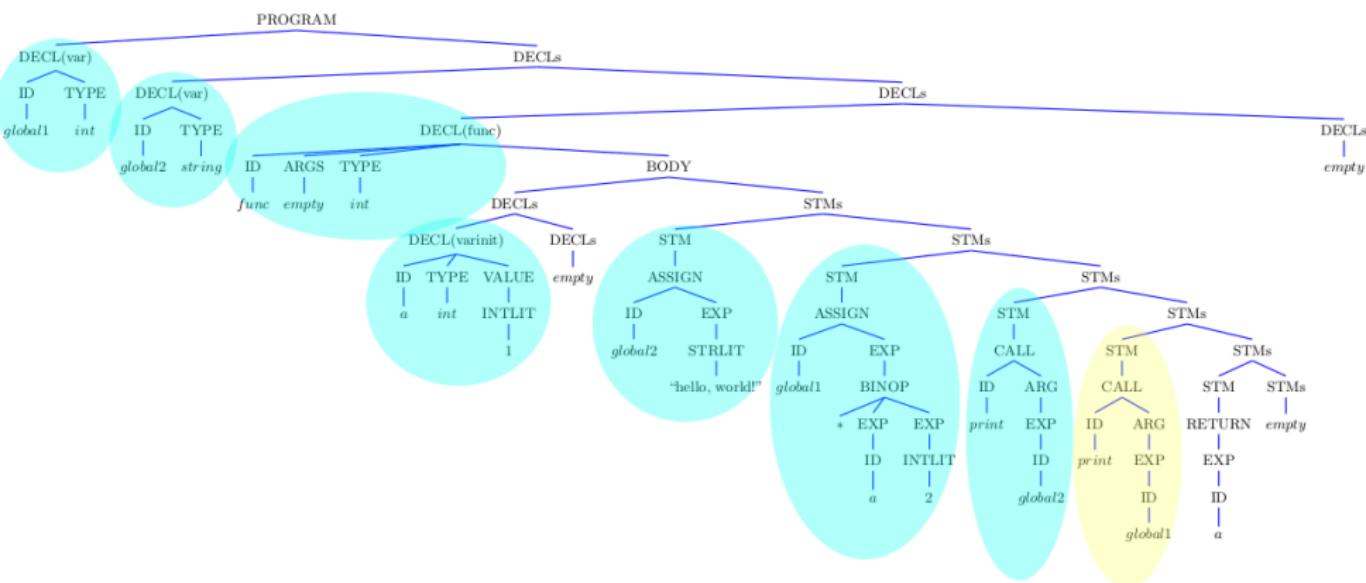
## SYMBOL TABLE - EXEMPLO



| ID      | TYPE   | ARGS   |
|---------|--------|--------|
| print   | void   | string |
| global1 | int    |        |
| global2 | string |        |
| func    | int    | void   |
| <hr/>   |        |        |
| a       | int    |        |

lookup(print)  
lookup(global2)

# SYMBOL TABLE - EXEMPLO



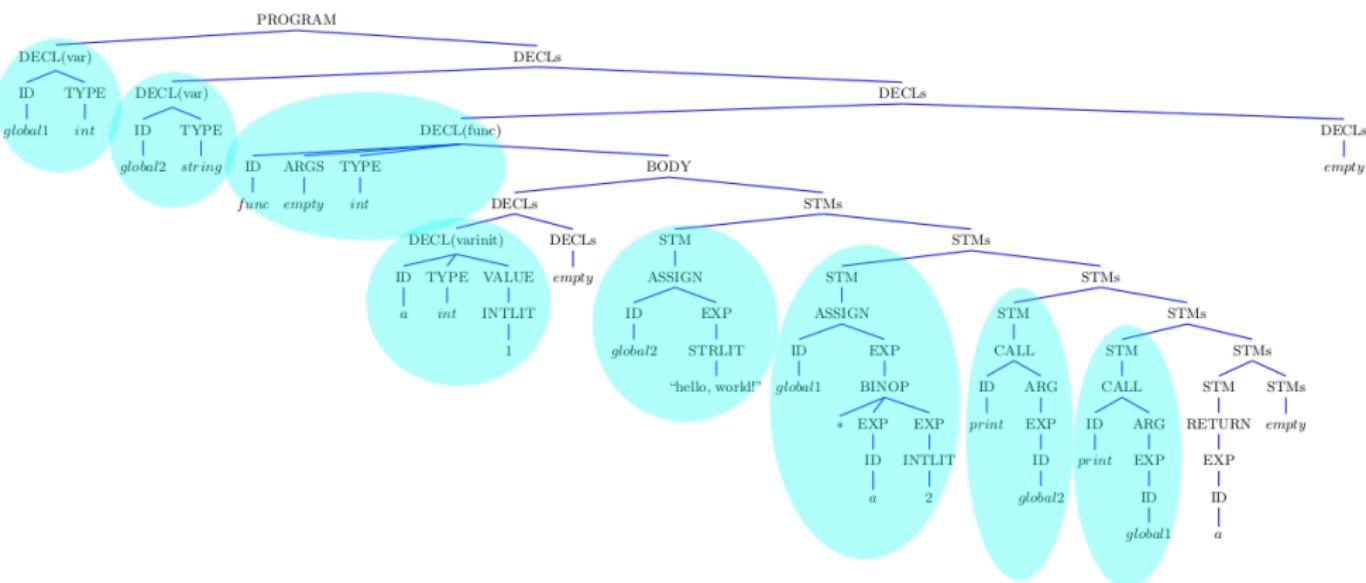
| ID      | TYPE   | ARGS   |
|---------|--------|--------|
| print   | void   | string |
| global1 | int    |        |
| global2 | string |        |
| func    | int    | void   |

---

|   |     |
|---|-----|
| a | int |
|---|-----|

lookup(print)  
lookup(global1)

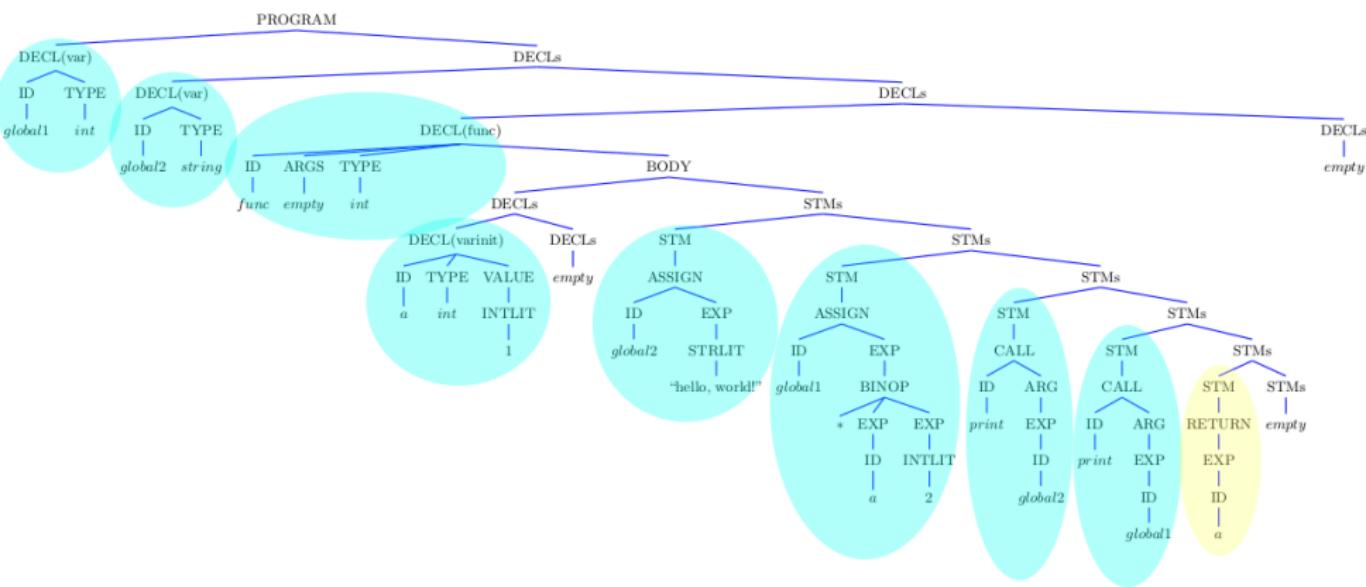
## SYMBOL TABLE - EXEMPLO



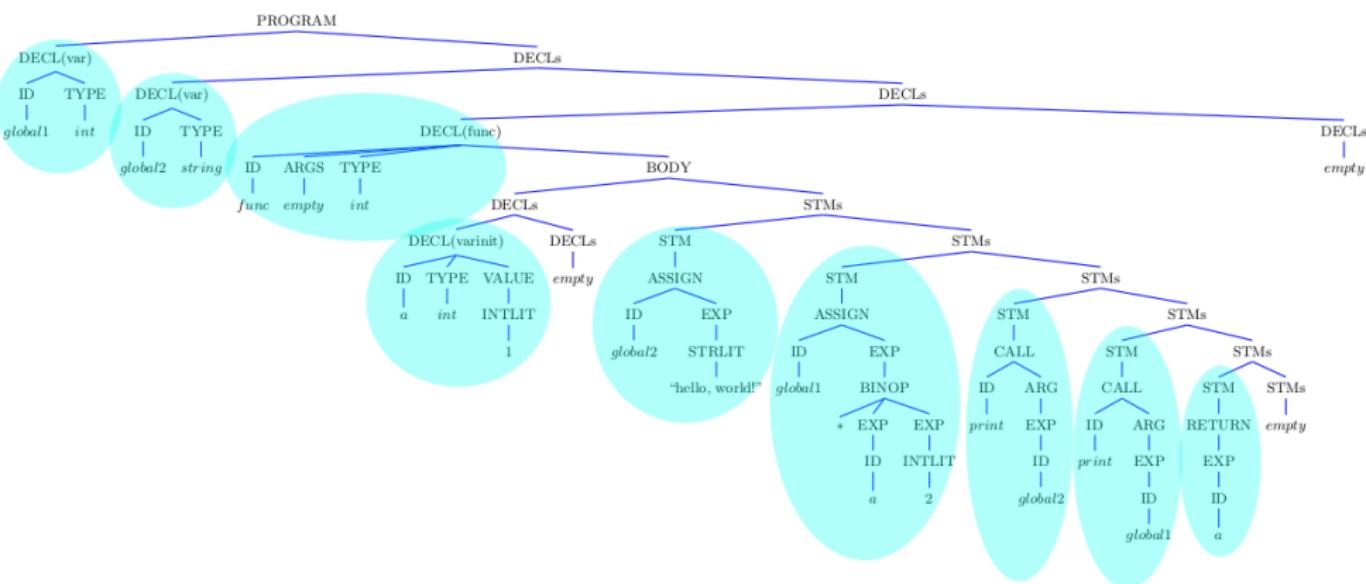
| ID      | TYPE   | ARGS |
|---------|--------|------|
| print   | void   | any  |
| global1 | int    |      |
| global2 | string |      |
| func    | int    | void |
| <hr/>   |        |      |
| a       | int    |      |

lookup(print)  
lookup(global1)

# SYMBOL TABLE - EXEMPLO



# SYMBOL TABLE - EXEMPLO



| ID      | TYPE   | ARGS |
|---------|--------|------|
| print   | void   | any  |
| global1 | int    |      |
| global2 | string |      |
| func    | int    | void |

---

|   |     |
|---|-----|
| a | int |
|---|-----|

lookup(a)  
lookup(func)