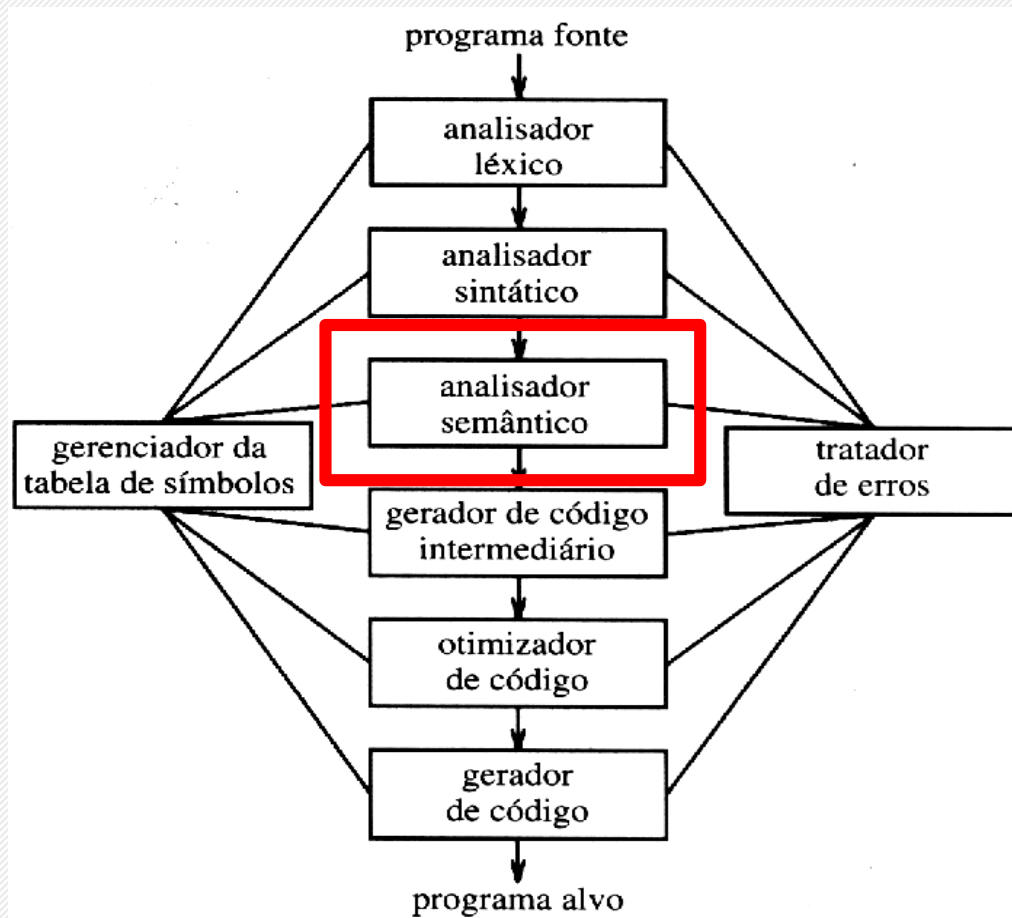


Análise semântica



Análise semântica



Análise semântica

Fundamentalmente, a análise semântica trata os aspectos sensíveis ao contexto da sintaxe das linguagens de programação.

Por exemplo, não é possível representar em uma gramática livre de contexto uma regra como:

"Todo identificador deve ser declarado antes de ser usado."

Análise semântica

Por exemplo, para o comando de atribuição **SOMA := SOMA + 35**, é necessário fazer a seguinte análise:

- O identificador SOMA foi declarado?
- O identificador SOMA é uma variável?
- Qual o escopo da declaração da variável SOMA: local ou global?
- Qual o tipo da variável SOMA? o valor atribuído no lado direito do comando de atribuição é compatível?

Análise semântica

Esse processo é implementado de forma dirigida pela sintaxe: associa-se a cada regra da gramática uma ação (ação semântica) a ser executada quando o analisador sintático sinaliza um uso da regra.

```
<decl> ::= <tipo> <listaid>  
<tipo> ::= inteiro #1  
          | real #1  
          | caracter #1  
          | logico #1  
<listaid> : <listaid> , id #2  
           | id #2
```

Análise semântica

Toda vez que o analisador sintático indicar o uso de uma regra associada a uma declaração, a rotina semântica associada a essa regra é chamada para acrescentar à tabela o identificador correspondente, fornecido pelo analisador léxico.

```
<decl> ::= <tipo> <listaid>  
<tipo> ::= inteiro #1  
          | real #1  
          | caracter #1  
          | logico #1  
<listaid> : <listaid> , id #2  
           | id #2
```



```
void Semantico::executeAction
```

Análise semântica

A ideia fundamental é a de usar tabelas externas ao processo de análise sintática, em que as informações são coletadas para posterior consulta.

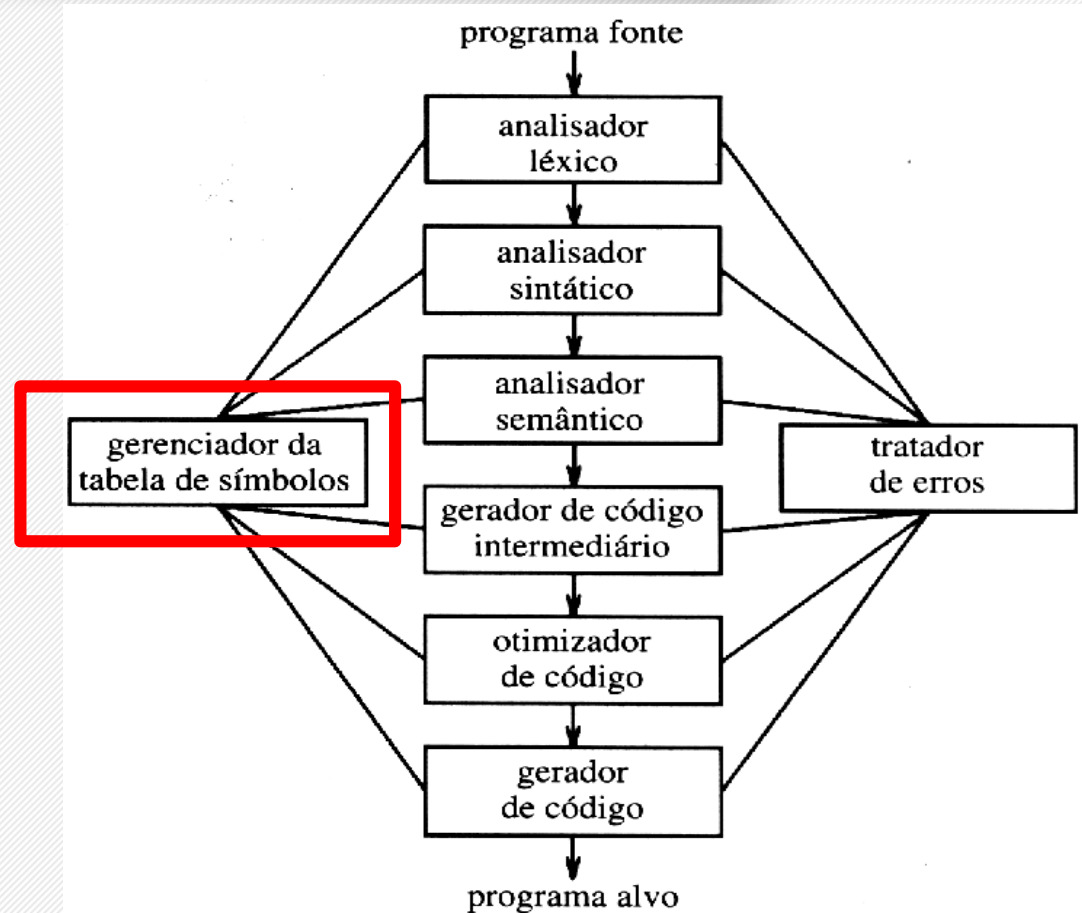


Tabela de Símbolos



Tabela de símbolos

- A tabela de símbolos serve como um banco de dados para o processo de compilação.
- Seu principal conteúdo são informações sobre tipos e atributos de cada nome definido pelo usuário no programa.
- Essas informações são colocadas na tabela de símbolos pelos analisadores léxico e sintático e usadas pelo analisador semântico e pelo gerador de código.

Tabela de símbolos

Inteiro a, b

Real vet [5]

Inteiro funcao coisa (inteiro a, real &x)

id	tipo	ini	usada	escopo	param	pos	vet	matriz	ref	func
a	Inteiro	F	F	global	F	0	F	F	F	F
b	Inteiro	F	F	global	F	0	F	F	F	F
vet	Real	F	F	global	F	0	T	F	F	F
coisa	Inteiro	F	F	global	F	0	F	F	F	T
a	Inteiro	F	F	coisa	T	1	F	F	F	F
x	Real	F	F	coisa	T	2	F	F	T	F

Tabela de símbolos

Inteiro a, b

Real vet [5]

Inteiro funcao coisa (inteiro a, real &x)

id	tipo	ini	usada	escopo	param	pos	vet	matriz	ref	func
a	Inteiro	F	F	global	F	0	F	F	F	F
b	Inteiro	F	F	global	F	0	F	F	F	F
vet	Real	F	F	global	F	0	T	F	F	F
coisa	Inteiro	F	F	global	F	0	F	F	F	T
a	Inteiro	F	F	coisa	T	1	F	F	F	F
x	Real	F	F	coisa	T	2	F	F	T	F

Tabela de símbolos

Inteiro a, b

Real vet [5]

Inteiro funcao coisa (inteiro a, real &x)

id	tipo	ini	usada	escopo	param	pos	vet	matriz	ref	func
a	Inteiro	F	F	global	F	0	F	F	F	F
b	Inteiro	F	F	global	F	0	F	F	F	F
vet	Real	F	F	global	F	0	T	F	F	F
coisa	Inteiro	F	F	global	F	0	F	F	F	T
a	Inteiro	F	F	coisa	T	1	F	F	F	F
x	Real	F	F	coisa	T	2	F	F	T	F

Tabela de símbolos

Inteiro a, b
Real vet [5]

Inteiro funcao coisa (inteiro a, real &x)

id	tipo	ini	usada	escopo	param	pos	vet	matriz	ref	func
a	Inteiro	F	F	global	F	0	F	F	F	F
b	Inteiro	F	F	global	F	0	F	F	F	F
vet	Real	F	F	global	F	0	T	F	F	F
coisa	Inteiro	F	F	global	F	0	F	F	F	T
a	Inteiro	F	F	coisa	T	1	F	F	F	F
x	Real	F	F	coisa	T	2	F	F	T	F

Tabela de símbolos

Inteiro a, b

Real vet [5]

Inteiro funcao coisa (inteiro a, real &x)

id	tipo	ini	usada	escopo	param	pos	vet	matriz	ref	func
a	Inteiro	F	F	global	F	0	F	F	F	F
b	Inteiro	F	F	global	F	0	F	F	F	F
vet	Real	F	F	global	F	0	T	F	F	F
coisa	Inteiro	F	F	global	F	0	F	F	F	T
a	Inteiro	F	F	coisa	T	1	F	F	F	F
x	Real	F	F	coisa	T	2	F	F	T	F

Tabela de símbolos

Como implementar a tabela de símbolos?

`List<Symbol>`

O que representa um símbolo?

Atributos da tabela

id	tipo	ini	usada	escopo	param	pos	vet	matriz	ref	func
String	String/ enum	bool	bool	String/ int	bool	int	bool	bool	bool	bool

Tabela de símbolos

Ações semânticas para Geração da tabela de Símbolos

```
<decl> ::= <tipo> <listaíd>
<tipo> ::= inteiro #1 -> executeAction(1,...)
          | real #1
          | caracter #1
          | logico #1
<listaíd> : <listaíd> , id #2 -> executeAction(2,...)
          | id #2
```

Semântico.java do GALS

```
void Semantico::executeAction (int action, const Token *token)...{
    ....
    switch (action) {
        case 1:.....
            break;
        case 2: .....
            break;
```


Tabela de símbolos

Ações semânticas para Geração da tabela de Símbolos

```
<decl> ::= <tipo> <listaid>  
<tipo> ::= inteiro #1  
          | real #1  
          | caracter #1  
          | logico #1  
<listaid> : <listaid> , id #2  
          | id #2
```

```
#1 tipo = token->getLexeme();  
#2 nome = token->getLexeme();  
    insere_tabela (tipo, nome);
```

Exemplo 1

```
inteiro a
```

Ação #1: Registra o tipo

Ação #2: Registra o id e insere na tabela com a informação do tipo

Tabela de símbolos

inteiro

Ações semânticas para Geração da tabela de Símbolos

```
<decl> ::= <tipo> <listaid>  
<tipo> ::= inteiro #1  
          | real #1  
          | caracter #1  
          | logico #1  
<listaid> : <listaid> , id #2  
          | id #2
```

```
#1 tipo = token->getLexeme();  
#2 nome = token->getLexeme();  
    insere_tabela (tipo, nome);
```

Exemplo 1

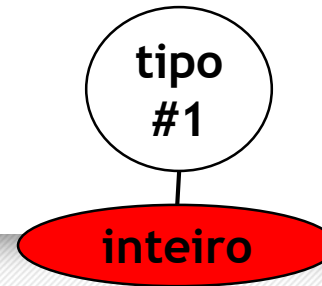
inteiro a

inteiro

Ação #1: Registra o tipo

Ação #2: Registra o id e insere na tabela com a informação do tipo

Tabela de símbolos



Ações semânticas para Geração da tabela de Símbolos

```
<decl> ::= <tipo> <listaid>  
<tipo> ::= inteiro #1  
          | real #1  
          | caracter #1  
          | logico #1  
<listaid> : <listaid> , id #2  
          | id #2
```

```
#1 tipo = token->getLexeme();  
#2 nome = token->getLexeme();  
    insere_tabela (tipo, nome);
```

Exemplo 1

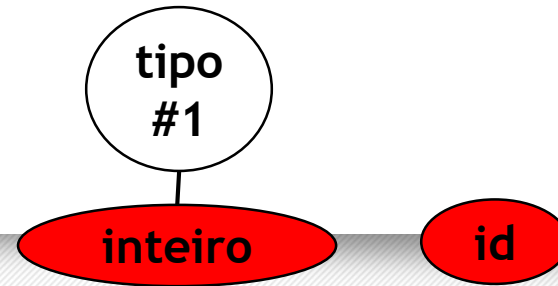
inteiro a

inteiro **#1** → tipo

Ação **#1**: Registra o tipo

Ação **#2**: Registra o id e insere na tabela com a informação do tipo

Tabela de símbolos



Ações semânticas para Geração da tabela de Símbolos

```
<decl> ::= <tipo> <listaíd>  
<tipo> ::= inteiro #1  
          | real #1  
          | caracter #1  
          | logico #1  
<listaíd> : <listaíd> , id #2  
           | id #2
```

```
#1 tipo = token->getLexeme();  
#2 nome = token->getLexeme();  
insere_tabela (tipo, nome);
```

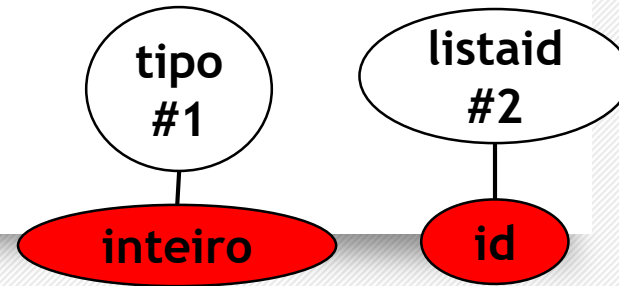
Exemplo 1



Ação #1: Registra o tipo

Ação #2: Registra o id e insere na tabela com a informação do tipo

Tabela de símbolos



Ações semânticas para Geração da tabela de Símbolos

```
<decl> ::= <tipo> <listaíd>  
<tipo> ::= inteiro #1  
         | real #1  
         | caracter #1  
         | logico #1  
<listaíd> : <listaíd> , id #2  
           | id #2
```

```
#1 tipo = token->getLexeme();  
#2 nome = token->getLexeme();  
   insere_tabela (tipo, nome);
```

Exemplo 1

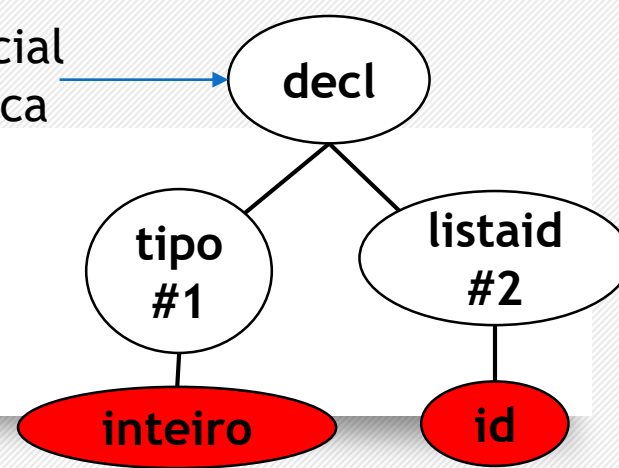


Ação **#1**: Registra o tipo

Ação **#2**: Registra o id e insere na tabela com a informação do tipo

Tabela de símbolos

Símbolo inicial
da gramática



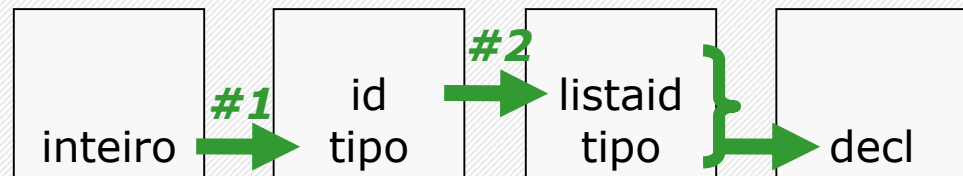
Ações semânticas para Geração da tabela de Símbolos

```
<decl> ::= <tipo> <listaid>  
<tipo> ::= inteiro #1  
          | real #1  
          | caracter #1  
          | logico #1  
<listaid> : <listaid> , id #2  
           | id #2
```

```
#1 tipo = token->getLexeme();  
#2 nome = token->getLexeme();  
    insere_tabela (tipo, nome);
```

Exemplo 1

inteiro a



Ação #1: Registra o tipo

Ação #2: Registra o id e insere na tabela com a informação do tipo

Tabela de símbolos

Ações semânticas para Geração da tabela de Símbolos

```
<decl> ::= <tipo> <listaíd>
<tipo> ::= inteiro #1
        | real #1
        | caracter #1
        | logico #1
<listaíd> : <listaíd> , id #2
          | id #2
```

```
#1 tipo = token->getLexeme();
#2 nome = token->getLexeme();
   insere_tabela (tipo, nome);
```

Exemplo 2

real b, c



Ação #1: Registra o tipo

Ação #2: Registra o id e insere na tabela com a informação do tipo

Tabela de símbolos

A **verificação de unicidade** detecta situações tais como duplicação em declarações de variáveis, de componentes de estruturas e em rótulos do programa.

```
#1 tipo = token->getLexeme();  
#2 nome = token->getLexeme();  
insere_tabela (tipo, nome); // verificar unicidade
```


Tabela de símbolos

Adicione ações semânticas na sua gramática para identificar e adicionar a tabela de símbolos

- Identificadores:
 - Variáveis (escalar e vetorial)
 - Sub-rotinas
 - Parâmetros de rotinas
 - Programa (ou função principal)