

ANÁLISE SEMÂNTICA

SUMÁRIO

- ✓ Introdução à análise semântica
- ✓ Tabela de símbolos
- ✓ Exemplos de ações para análise semântica da LMS

SUMÁRIO

- ✓ **Introdução à análise semântica**
- ✓ Tabela de símbolos
- ✓ Exemplos de ações para análise semântica da LMS

INTRODUÇÃO À ANÁLISE SEMÂNTICA

- ✓ A análise semântica garante que o programa fonte satisfaça **regras adicionais** em relação à linguagem de programação definida
- ✓ Exemplos de regras semânticas:
 - ✓ Uma variável precisa ser declarada antes de ser usada
 - ✓ Uma variável pode ser declarada somente uma vez no mesmo escopo
 - ✓ Em uma operação de atribuição, o tipo de dado utilizado deve ser condizente com o tipo da variável declarada
 - ✓ A condição de uma declaração utilizada em um “if” deve ser do tipo booleana

INTRODUÇÃO À ANÁLISE SEMÂNTICA

- ✓ Durante a análise semântica são extraídas informações do programa-fonte úteis à geração de código
- ✓ As regras semânticas podem ser agrupadas em (exemplos):

✓ Estáticas x Dinâmicas



Mais ampla

✓ Tipo x Escopo

INTRODUÇÃO À ANÁLISE SEMÂNTICA

- ✓ Regras de verificações estáticas
 - ✓ Controle de fluxo (escopo...)
 - ✓ Singularidade (declarações únicas de ident...)
 - ✓ Tipo (verificações de tipos...)
- ✓ Regras de verificações dinâmicas
 - ✓ Limites de vetores
 - ✓ Acesso a ponteiros nulos

Análise semântica

INTRODUÇÃO À ANÁLISE SEMÂNTICA

✓ Regras de Tipo x Escopo

REGRAS DE TIPO

Classificam as construções em categorias, que possuem regras específicas para uso

- Atribuições →	<code>i = 0;</code>	<code>void</code>
- Expressões →	<code>i == 1</code>	<code>boolean</code>
- Funções →	<code>int soma(int a,int b);</code>	<code>int x int → int</code>

REGRAS DE TIPO - EXEMPLOS

✓ Exemplos:

- ✓ O **valor atribuído** a uma variável precisa ser do mesmo tipo que a variável foi declarada
- ✓ Os **operandos de expressões aritméticas** (*, ÷, +, -) precisam ser do tipo inteiro
- ✓ Os **operandos de comparações** (==, !=) precisam ser do tipo inteiro ou real
- ✓ Os **tipos dos argumentos** de métodos devem coincidir com os respectivos tipos declarados no cabeçalho do método

REGRAS DE ESCOPO

- ✓ Declaração de identificadores e as respectivas regiões em que seu uso é permitido
- ✓ Exemplos de escopo
 - ✓ Bloco de declaração
 - ✓ Corpo de um método
 - ✓ Programa inteiro

ESCOPO - EXEMPLOS

```
{  
    int a;  
    {  
        int b;  
    }  
}
```

The diagram illustrates variable scope using nested blocks and dashed lines. The outermost block is defined by a large pair of curly braces. Inside this block, the variable `int a;` is declared. Within the scope of `a`, there is a nested block defined by a smaller pair of curly braces. Inside this nested block, the variable `int b;` is declared. Dashed lines connect the opening and closing braces of each block to show the extent of their respective scopes. The scope of `a` covers the entire content of the outer block, while the scope of `b` is limited to the inner block.

ESCOPO - EXEMPLOS

```
int soma (int a,int b) {  
    int x;  
    if (a==b) {  
        int z;  
        ...  
    }  
    ...  
}
```

ESCOPO - BURACOS DE ESCOPO

```
double x;
```

```
int print (int x) {
```

```
    int abc;
```

```
    x = 4;
```

```
}
```

```
x = 3.5;
```

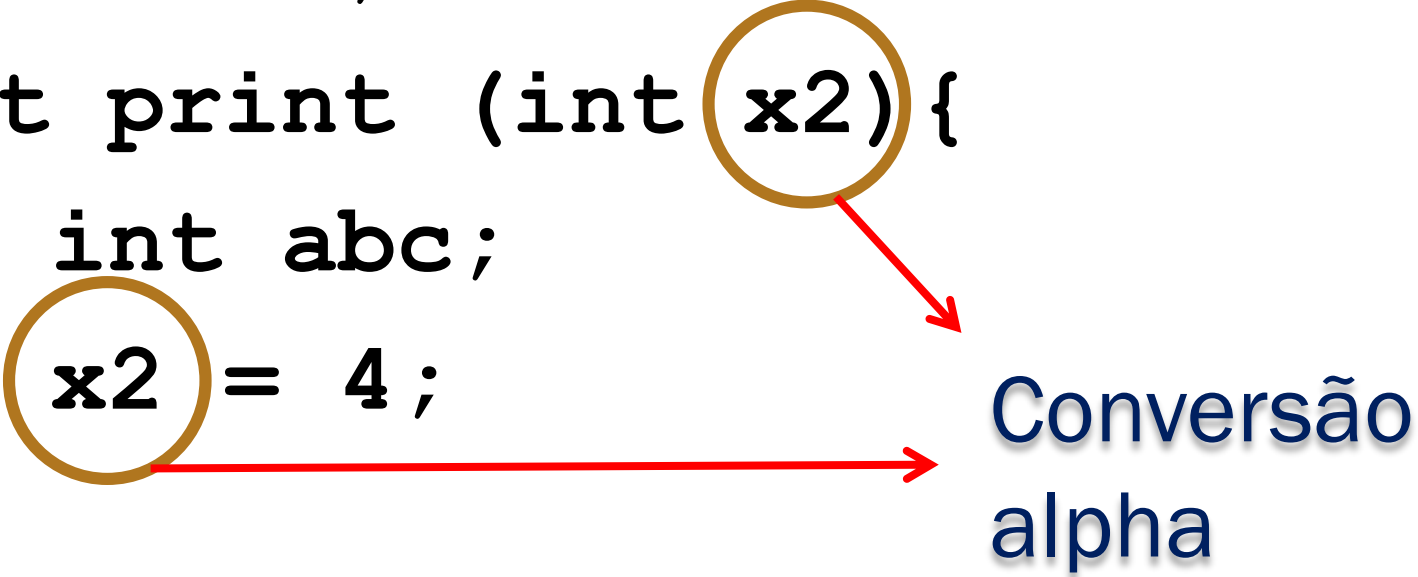
```
...
```

Buraco de
escopo
para
double x

ESCOPO - CONVERSÃO ALPHA

```
double x;  
int print (int x2) {  
    int abc;  
    x2 = 4;  
}  
x = 3.5;  
...
```

Conversão
alpha



ESCOPO - CONVERSÃO ALPHA

- ✓ Algumas otimizações de código tornam-se mais fáceis evitando-se buracos de escopo
- ✓ Os compiladores podem incluir uma etapa em que todas as variáveis sejam renomeadas

REGRAS DE ESCOPO - EXEMPLOS

- ✓ Um identificador somente pode ser usado no escopo em que foi declarado
- ✓ Não é permitido que sejam declarados dois ou mais identificadores com o mesmo nome, no mesmo escopo

SUMÁRIO

- ✓ Introdução à análise semântica
- ✓ **Tabela de símbolos**
- ✓ Exemplos de ações para análise semântica da LMS

TABELA DE SÍMBOLOS (TS)

- ✓ Usada para controlar informações a respeito de nomes
- ✓ Quando um **nome é encontrado**, uma **busca** é efetuada na TS
- ✓ Quando um **novo nome** é encontrado ou alguma **nova informação** a respeito de um nome é encontrada, uma **alteração** é efetuada na TS

TABELA DE SÍMBOLOS (TS)

- ✓ Usada para verificações semânticas relacionadas a tipos, escopos
- ✓ Também utilizada para armazenar informações sobre identificadores que serão úteis na geração de código

TABELA DE SÍMBOLOS - EXEMPLO

	Nome	Categoria	Nível	Dado 1	Dado 2
	abc	Variável	0		
	soma	Procedure	0		
	a	Parâmetro	0		
	i	Variável	1		

TABELA DE SÍMBOLOS

- ✓ Não se constrói durante a análise léxica, pois a hierarquia de escopos é definida na sintaxe
- ✓ Construída durante a análise sintática, através do uso de ações semânticas

ESTRUTURAS DE DADOS COMUNS

✓ Listas lineares:

- ✓ Vantagem: implementação simples
- ✓ Desvantagem: acesso custoso

✓ Pilha

- ✓ Vantagem: símbolos declarados por último são alcançados primeiro
- ✓ Desvantagem: é necessário desempilhar os elementos até chegar ao símbolo desejado

✓ Hashing

- ✓ Vantagem: acesso rápido
- ✓ Desvantagem: se torna lento se houverem muitas colisões

HASHING – ESQUEMA BÁSICO


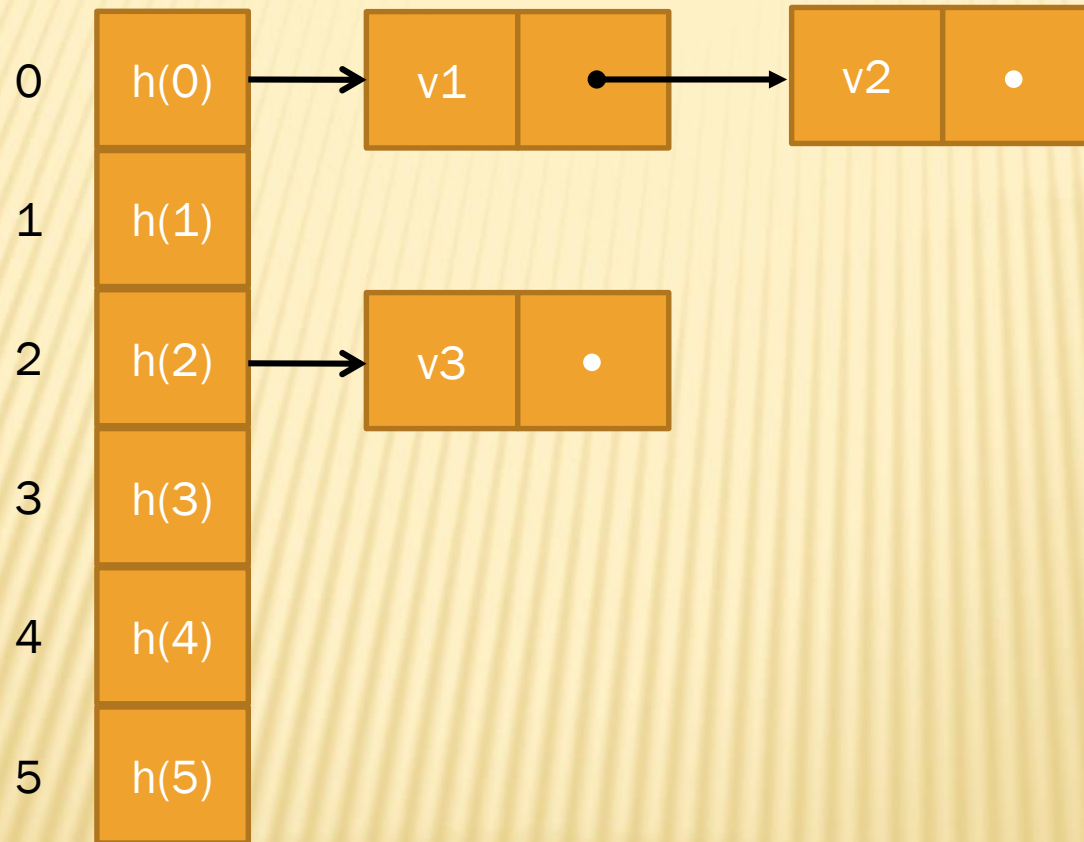
- ✓ Tabela hash, consistindo basicamente de um vetor
- ✓ Entradas, normalmente organizadas em listas encadeadas
- ✓ Função hash muito comum: $h(k) = k \% m$

resto da divisão inteira

Tabela de símbolos → Estruturas de dados

HASHING – ESQUEMA BÁSICO



FUNÇÃO HASH - EXEMPLO

$$\text{ind} = \sum_{i=0}^{cmp-1} C_{cmp-1-i} \cdot A^i$$

cmp → comprimento da string

A → valor base do polinômio

índice final = ind % m

FUNÇÃO HASH - EXEMPLO

✓ Exemplo: “casa” | A=37

$$\text{índice} = 'a'.37^0 + 's'.37^1 + 'a'.37^2 + 'c'.37^3$$

✓ Calcular o polinômio usando o método de Horner

✓ Pode-se usar a notação convencional, com o primeiro caractere iniciando pelo índice 0, ou seja:

$$\text{índice} = 'c'.37^0 + 'a'.37^1 + 's'.37^2 + 'a'.37^3$$

FUNÇÃO HASH - EXEMPLO

Método de Horner

$$\begin{aligned} a_0 + a_1x + a_2x^2 + \cdots + a_nx^n &= a_0 + x(a_1 + a_2x + \cdots + a_nx^{n-1}) \\ &= a_0 + x(a_1 + x(a_2 + a_3x + \cdots + a_nx^{n-2})) \end{aligned}$$

Exemplo: “casa”

$$\text{índice} = 'c'.37^0 + 'a'.37^1 + 's'.37^2 + 'a'.37^3$$



Método de Horner

$$\text{índice} = 'c' + 37('a' + 37('s' + 37('a')))$$

ANÁLISE SEMÂNTICA

AÇÕES SEMÂNTICAS

EXEMPLOS PARA A LMS

Análise semântica → Ações semânticas

EXEMPLOS DE AÇÕES PARA A LMS

<PROG> ::= PROGRAM ID #100 ";" <BLOCO> "." #101

Resumo:

#100 → Inicialização de variáveis de controle utilizadas durante toda a análise semântica...

#101 → Gera instrução de encerramento do programa...

EXEMPLOS DE AÇÕES PARA A LMS

<DCLVAR> ::= ε

<DCLVAR> ::= VAR #107 <LID> ":" <TIPO> ";" <LDVAR>

<LID> ::= ID #104 <REPID>

<REPID> ::= ε

<REPID> ::= "," ID #104 <REPID>

<TIPO> ::= INTEGER

<LDVAR> ::= ε

<LDVAR> ::= <LID> ":" <TIPO> ";" <LDVAR>

	Nome	Categoria	...
	abc	Variável	...

#107 → Seta tipo de identificador = VARIÁVEL

#104 → Insere nome na tabela de símbolos...

Análise semântica → Ações semânticas

EXEMPLOS DE AÇÕES PARA A LMS

<COMANDO> ::= ID #114 "[:=" <EXPRESSAO> #115

#114 → Verifica se o nome está na tabela de símbolos e se é do tipo VARIÁVEL...

#115 → Gera instrução para armazenamento do valor na variável

Análise semântica → Ações semânticas

EXEMPLOS DE AÇÕES PARA A LMS

$\langle \text{COMANDO} \rangle ::= \text{IF} \langle \text{EXPRESSAO} \rangle \#120 \text{ THEN} \langle \text{COMANDO} \rangle \langle \text{ELSEPARTE} \rangle \#121$

$\langle \text{ELSEPARTE} \rangle ::= \varepsilon$

$\langle \text{ELSEPARTE} \rangle ::= \#122 \text{ ELSE } \langle \text{COMANDO} \rangle$

#120 → Gera instrução de “desvia se falso” e empilha endereço para resolução posterior...

#122 → Resolve endereço de desvio do IF que está na pilha; Gera instrução “desvia sempre” e empilha endereço para resolução posterior...

#121 → Completa instrução “desvia sempre” gerada em “#122”

Análise semântica → Ações semânticas

EXEMPLOS DE AÇÕES PARA A LMS

<COMANDO> ::= CALL ID #116 <PARAMETROS> #117

#116 → Consulta a tabela de símbolos e salva o endereço do nome...

#117 → Verifica a quantidade de parâmetros informada e gera instrução da chamada de procedure usando endereço salvo na tabela de símbolos...