

ANÁLISE SEMÂNTICA E GERAÇÃO DE CÓDIGO

FUNÇÃO



Determinar o significado de cada construção sintática e então **traduzir o programa** escrito na linguagem **fonte para o código intermediário** da máquina virtual. Na determinação do significado do programa fonte, o analisador semântico:

- verifica a coerência da declaração e do uso de identificadores;
- cria e mantém a tabela de símbolos;
- detecta e trata erros semânticos.

ESPECIFICAÇÃO DA SEMÂNTICA DE UMA LINGUAGEM DE PROGRAMAÇÃO: "ao contrário da sintaxe, que é facilmente formalizável, (...) a semântica, apesar de também poder ser expressa formalmente, exige para isto notações substancialmente mais complexas, de aprendizagem mais difícil" (JOSÉ NETO, 1987)

- **informal:** através de textos em linguagem natural
- **semi-formal:** ações semânticas embutidas na gramática
- **formal:** gramática de atributos, semântica de ações, semântica denotacional, semântica operacional, semântica axiomática.

IMPLEMENTAÇÃO: usar a **tradução dirigida pela sintaxe** onde a **geração do código-intermediário** normalmente é feita pelo conjunto de rotinas, as **ações semânticas**, que são embutidas na gramática através de símbolos e indicam verificações de ordem semântica a serem efetuadas e instruções a serem geradas no processo de compilação.

MÁQUINA VIRTUAL

No projeto do analisador semântico e gerador de código, deve-se decidir se alguma **representação intermediária** será usada ou se será gerado **código de máquina**.

FORMAS de representação intermediária $\left\{ \begin{array}{l} \text{notação polonesa (prefix e postfix)} \\ \text{código de 3 endereços} \\ \text{árvore} \end{array} \right.$

- **notação postfix** (arquitetura da máquina alvo = pilha):

EXEMPLOS: (transformações sintáticas da entrada)

$a + b * c$	$a b c * +$
$(a + b) * c$	$a b + c *$
$a := b * c + b * d$	$a b c * b d * + :=$

- **código de 3 endereços** (triplos e quádruplas):

EXEMPLOS:

1. $a := b * c + b * d$ (transformações sintáticas da entrada)

$(*, b, c)$	$(*, b, c, t1)$
$(*, b, d)$	$(*, b, d, t2)$
$(+, \{1\}, \{2\})$	$(+, t1, t2, t3)$
$(:=, \{3\}, a)$	$(:=, t3, a, -)$

2. $a := b * c + b * d$, sendo a, d variáveis reais e b, c variáveis inteiras (tradução baseada na semântica da linguagem)

(MULTI, addr(b), addr(c))
(FLOAT, addr(b), -)
(MULTF, addr(b), addr(d))
(FLOAT, {1}, -)
(ADDF, {4}, {3})
(:=, {5}, addr(a))

TABELA DE SÍMBOLOS

Armazena **informações** relativas aos **identificadores** encontrados no programa fonte durante o processo de compilação.

- **IMPORTÂNCIA:** a definição dos identificadores aparece em um único lugar no programa (declaração), enquanto o identificador pode ser usado em vários lugares.
- **ORGANIZAÇÃO:**

NOME	CATEGORIA pode ser	ATRIBUTO depende da categoria
identificador analisado	0: identificador de programa	nada
	1: identificador de variável inteira	deslocamento (endereço de memória) na pilha de dados
	2: identificador de variável real	deslocamento (endereço de memória)
	3: identificador de variável literal	deslocamento (endereço de memória)
	4: identificador de variável lógica	deslocamento (endereço de memória)

- **OPERAÇÕES:**
 - ✓ **criar:** usada antes de começar a análise semântica
 - ✓ **destruir:** usada após executar a análise semântica
 - ✓ **pesquisar:** usada quando da compilação do programa
 - ✓ **inserir:** quando da compilação da declaração de variáveis
 - ✓ **alterar:** usada quando na compilação da declaração de variáveis
 - ✓ **recuperar:** usada quando da compilação do corpo do programa
- **IMPLEMENTAÇÃO:** lista ordenada, árvores binárias de busca, dicionário, tabela *hash*, etc.

ESPECIFICAÇÃO DAS AÇÕES SEMÂNTICAS

- são a parte do compilador que **interpreta o significado de um programa** baseada na estrutura sintática;
- são associadas com regras de produção da gramática ou sub-árvores de uma árvore sintática;
- são **explicitamente projetadas** para cada construção da linguagem:

*<comando if> ® **if** <expressão> #StartIF **then** <lista comandos> **endif** #FinishIF*

- nem todos os símbolos sintáticos (terminais e não-terminais) têm ações semânticas associadas, sendo que algumas regras de produção podem incluir mais do que uma ação semântica;
- ações semânticas não chamam outras ações semânticas explicitamente, mas utilizam **registros semânticos** para trocar informações umas com as outras;
- o processamento pode incluir:
 - ✓ geração de código
 - ✓ registro de informações na tabela de símbolos
 - ✓ busca de identificadores e atributos associados na tabela de símbolos
 - ✓ verificação dos tipos dos argumentos para determinar o tipo do resultado
 - ✓ construção de registros semânticos
- as ações semânticas são tratadas como outros símbolos da gramática e colocados na pilha sintática juntamente com a regra de produção a qual está associada, devendo o **algoritmo de análise sintática ser alterado**.