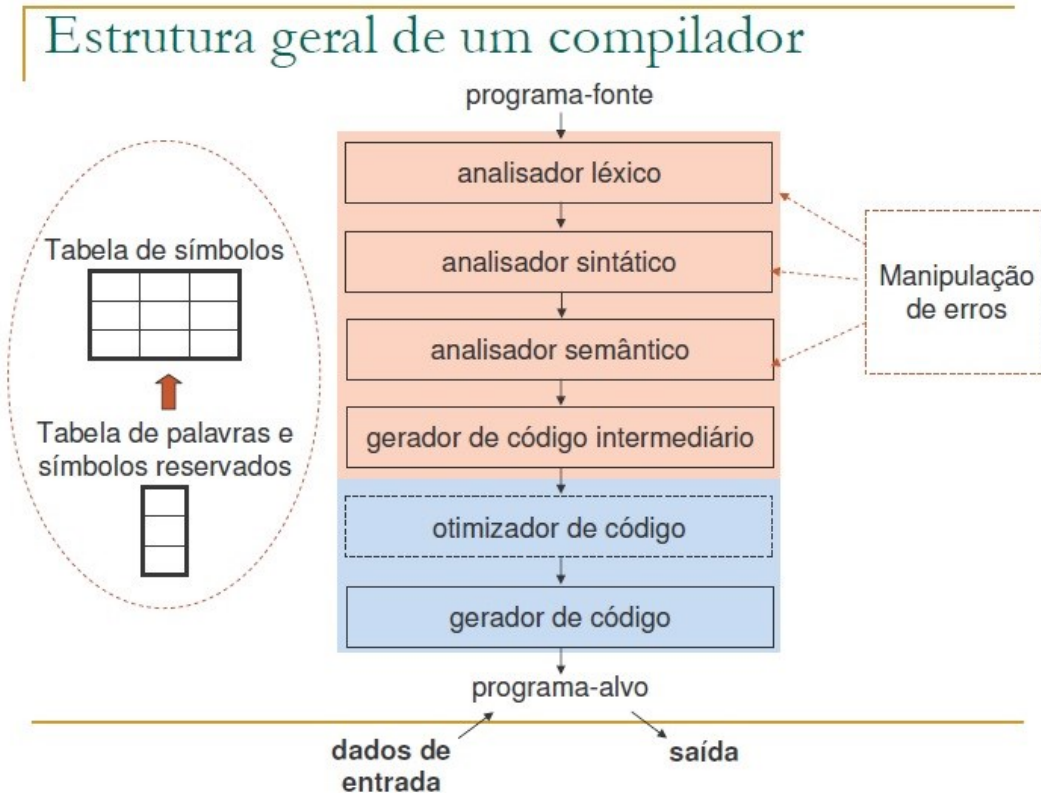


ANÁLISE SEMÂNTICA

A análise semântica é a terceira etapa do processo de compilação. Nela ocorre a validação de uma série de regras que não podem ser verificadas nas etapas anteriores.



Muitos erros em um código fonte não podem ser detectados sintaticamente, pois precisam de contexto. Por exemplo: verificar se todas as variáveis utilizadas foram declaradas, verificar se as atribuições realizadas nas variáveis estão de acordo com os tipos das variáveis.

A análise semântica é responsável por verificar aspectos relacionados ao significado das sentenças. As validações executadas durante a análise semântica possuem o objetivo de garantir que o programa fonte esteja de acordo com as regras da gramática da linguagem fonte.

A análise semântica percorre a árvore sintática e relaciona os identificadores com seus dependentes de acordo com a estrutura hierárquica.

A análise semântica também captura informações sobre o programa fonte para que as fases subsequentes possam gerar o código objeto.

Uma tarefa importante da análise semântica é a verificação de tipos, analisando se cada operador recebeu os operandos permitidos e especificados na linguagem fonte.

O analisador semântico utiliza a árvore sintática gerada pelo analisador sintático para realizar diversas tarefas. Algumas são citadas na figura abaixo.

Análise semântica

Análise semântica

O analisador semântico utiliza a árvore sintática para:

- identificar operadores e operandos das expressões,
- reconhecer erros semânticos,
- fazer verificações de compatibilidade de tipo,
- analisar o escopo das variáveis,
- fazer verificações de correspondência entre parâmetros atuais e formais.

A Análise Semântica envolve algumas verificações, como:

- Verificação de tipos.
- Verificação do fluxo de controle.
Exemplo: os comandos que fazem o fluxo de controle deixar uma construção precisam ter algum local para onde transferir o controle.
- Verificações de unicidade.
Exemplo: um identificador precisa ser declarado univocamente no mesmo nível.

REALIZAÇÃO DA ANÁLISE SEMÂNTICA

- **Estática**, é feita no processo de compilação: linguagens tipadas, que exigem declarações.
Exemplo: C, Pascal, entre outras.
- **Dinâmica**, é feita em tempo de execução: linguagens em que as variáveis são determinadas pelo contexto de uso.
Exemplo: LISP, PROLOG, PHP, entre outras.

DESCRIÇÃO SEMÂNTICA

Formas para descrever a semântica:

- Pode ser especificada informalmente (geralmente é artesanal). As descrições são realizadas nos manuais de cada linguagem.
- Pode ser especificada formalmente, usando Gramática de Atributos.

Gramática de atributos

- O processo ocorre junto com a análise sintática.
- São agregados atributos aos símbolos não- terminais da gramática.
- São atreladas ações semânticas às produções da gramática.
- As ações semânticas associadas às produções computam os valores dos atributos dos símbolos.
- Cada símbolo pode possuir um ou mais atributos associados.
- Os atributos podem ser de qualquer tipo: uma cadeia, um número, um tipo, uma localização de memória, etc.
- O valor de um atributo é definido por uma regra semântica associada a produção que contém o símbolo dono do atributo.
- Pode ocorrer de uma regra de produção não apresentar nenhuma regra semântica.
- Nem todo símbolo gramatical tem atributos.

Exemplo de gramática de atributos

```
exp → exp + termo | exp - termo | termo
termo → termo * fator | termo div fator | fator
fator → ( exp ) | num
```

Regras semânticas que manipulam os atributos

✓ Por exemplo, regra para somar os atributos valores de duas variáveis

✦ $x := a + b$, cuja regra é
 $x.val := a.val + b.val$

Regras gramaticais	Regras semânticas
$exp \rightarrow exp_1 + termo$	$exp.val = exp_1.val + termo.val$
$exp \rightarrow exp_2 - termo$	$exp.val = exp_2.val - termo.val$
$exp \rightarrow termo$	$exp.val = termo.val$
$termo \rightarrow termo_1 * fator$	$termo.val = termo_1.val * fator.val$
$termo \rightarrow termo_2 \text{ div } fator$	$termo.val = termo_2.val / fator.val$
$termo \rightarrow fator$	$termo.val = fator.val$
$fator \rightarrow (exp)$	$fator.val = (exp.val)$
$fator \rightarrow num$	$fator.val = num.val$

ANÁLISE SEMÂNTICA COM CUP

Além de ser utilizado no desenvolvimento dos analisadores sintáticos de compiladores, o CUP também pode ser utilizado para a construção do analisador semântico. O CUP permite ao desenvolvedor associar ações semânticas às regras de produção previamente especificadas na etapa de análise sintática. Essas ações semânticas, no caso do CUP, nada mais são de que trechos de código Java onde o desenvolvedor, utilizando as características da linguagem fonte, define as ações semânticas. Um exemplo de ação semântica corresponde ao tratamento de erros semânticos (verificação de utilização de variáveis que não foram declaradas, verificação de tipos, verificação de escopo de variáveis, entre outros). A linguagem Java não possui construtores específicos para o escopo de analisadores semânticos.

Para essa etapa de análise o CUP não fornece ao programador uma API para manipulação da tabela de símbolos, assim como não fornece uma maneira padronizada para o tratamento de erros e conflitos inerentes à fase de análise semântica.