

# Análise Semântica

Percorre a **Árvore Sintática** para validar regras semânticas.

Avaliam a relação entre tokens distantes entre si.

Utiliza informações armazenadas na **Tabela de Símbolos**.

# Análise Semântica - Tabela de Símbolos

Estrutura de dados contendo um registro para cada **identificador** usado no programa.

Cada registro contém informações sobre seus **classificação, tipos, escopo** etc.

# Tabela de Símbolos - Exemplo

## Regras

- Os identificadores precisam ser declarados antes de usados.
- Quantidade e tipos de parâmetros da função igual na chamada.

Para **todo ID**: Nome, classificação, tipo e escopo.

Para **ID Função**: Quantidade de parâmetros.

Para **ID Parâmetro**: Ordem na lista.

# Tabela de Símbolos - Exemplo

```
void Fatorial ( int x ) {  
    if ( x > 1 ) return x * fatorial( x-1 );  
    return 1;  
}
```

<b>Nome</b>	<b>classificação</b>	<b>tipo</b>	<b>escopo</b>	<b>Qtd</b>	<b>ordem</b>
Fatorial	função	void	Global	1	-
x	parâmetro	int	Fatorial	-	1°

# Tabela de Símbolos - Formato

Estrutura influencia na velocidade de acesso a espaço utilizado.

Deve permitir uma consulta rápida aos registros.

Quais informações depende das regras semânticas.

# Análise Semântica - Regras

As regras semânticas são ambiguidades que surgirão na fase de síntese.

Decisão de como resolver deve ser explicitada pelo programador.

Regras que não conseguem ser descritas como regras sintáticas.

# Análise Semântica - Tipos de Regras

Compatibilizar tipos (atribuições, expressões etc...)

Analizar escopo de identificadores (variáveis, funções etc...)

Quantidade e tipos de argumentos na chamada de funções.

# Análise Semântica - Exemplos de Regras

O identificador deve ser declarado antes de utilização.

Somente identificadores **variável** podem aparecer do lado esquerdo de uma **atribuição**.

O tipo do **argumento** deve ser compatível com o tipo do **parâmetro**.

# Análise Semântica - Especificação

- **Informal:** Descrição no manual da linguagem (artesanal).
- **Formal:**
  - Vienna Definition Language (VDL);
  - Definições Axiomáticas;
  - Modelos Denotacionais;
  - Modelos Operacionais;
  - Gramática de Atributos.

# Análise Semântica - Gramática de Atributos

Gramática LC associada contendo regras sobre atributos dos identificadores.

Necessita que sejam armazenados os atributos relacionados aos identificadores.

Quais atributos serão armazenados varia conforme as regras necessitarem.

O armazenamento é feito durante a análise sintática.

## **Local de armazenar atributos**

**Interno:** Árvore sintática.

**Externo:** Tabela de símbolos.

# Análise Semântica - Erros

Toda regra semântica, de forma geral, pode ser substituídas por uma tomada de decisão automática do compilador nas fases de síntese.

Exemplo:

**Ambigüidade:** Tipos diferentes em operação aritmética.

**Decisão automática:** Type cast para o tipo mais restritivo.

# Análise Semântica - Algoritmo

As regras semânticas estão associadas a um ou mais **Não-Terminais**.

Um **Não-Terminal** pode ter várias regras associadas.

Percorrer a árvore e validar regras associadas ao visitar cada não-terminal.

# Análise Semântica - Algoritmo

```
função semantico( No n ){
    para todos t.filhos é igual c faça
        semantico( filho );
    atualizarTabelaSimbolo( tab, n );
    validarRegra( n );
}
```

# Algoritmo - Tabela de Símbolos

Preenchida com informações em sub-árvores **Declaração**.

- Inserir** entrada ao encontrar ID.
- Atualizar** entrada com informações usadas nas regras.

Consultada ao validar **Regras Semânticas**

- Buscar** entrada ao encontrar nó com regras associadas.

# Análise Semântica - Algoritmo

```
função atualizarTabelaSimbolo( Tabela tab, No n ){
    se em sub-árvore DECLARAÇÃO então
        se n é igual ID então
            adicionar( tab, n );
        senão
            atualizar( tab, n );
    }
}
```