

# Novas Regras Semânticas Baseadas no BNF

## Regras Semânticas Adicionais

### 1. Tipos de Variáveis e Coerência Semântica:

- Validar que variáveis do tipo "intero", "floati" e "string" sejam consistentes com os valores atribuídos.
- Exemplo: intero x = 10; (válido), string nome = 20; (inválido).

### 2. Escopo de Blocos:

- Garantir que variáveis declaradas dentro de um "Block" não sejam acessíveis fora dele.
- Validar que identificadores sejam únicos dentro do mesmo escopo.

### 3. Estruturas Condicionais e Controle de Fluxo:

- Validar que a condição de "Uai" ou "Quanto" seja uma expressão lógica válida.
- Bloquear ciclos infinitos não controlados.

### 4. Declaração de Ponteiros:

- Garantir que apenas tipos válidos (intero, floati, string) podem ser usados como ponteiros.
- Exemplo: intero \*ptr = &x; (válido).

### 5. Operadores:

- Validar que os operadores aritméticos (+, -, \*, /) sejam usados com tipos numéricos.
- Operadores lógicos (>, <, ==, !=) devem operar com tipos compatíveis.

### 6. Estruturas de Dados:

- Vetores ("TREM") devem ter tamanhos positivos e valores compatíveis com o tipo declarado.

## Novas Regras Semânticas Baseadas no BNF

- Exemplo: `intero arr[10] = [1, 2, 3];` (válido).

### 7. Funções:

- Garantir que parâmetros passados correspondam aos tipos esperados.
- Validar retorno compatível com o tipo definido.

### 8. Comentários e Formatação:

- Bloquear comentários aninhados ou não encerrados corretamente.
- Exemplo: `/* válido */` inválido.

### 9. Declarações e Atribuições:

- Verificar que as constantes declaradas não sejam alteradas em tempo de execução.
- Exemplo: `const intero max = 100; max = 200;` (inválido).

### 10. Estruturas Heterogêneas ("Istrut"):

- Validar que o número de elementos seja consistente com o tamanho declarado.
- Exemplo: `Istrut data[3] = {"A", 1, 2.0};` (válido).