

## Parte III: Analisador Semântico

**Sílvio Bandeira**

Professor Adjunto

### 1. Introdução

Complemente seu *parser* com um analisador semântico. Como a linguagem é simples, haverá basicamente checagem de tipos.

Sugere-se que o sintático não seja um código à parte, mas que esteja embutido no *parser*.

**Observação 1:** o arquivo a ser compilado será passado ao seu compilador **via argumento da linha de comando**

**Observação 2:** Imprimir **apenas mensagens de erro**.

**Observação 3:** A mensagem deve ser **clara e específica de erro, sempre que for o caso, e em qualquer fase do compilador. Formato: "ERRO na linha *n*, coluna *m*, ultimo token lido *t*: mensagem específica do erro"**

### 2. Regras

Qualquer comando que relacionar duas ou mais entidades (como variáveis e constantes) deverá verificar a compatibilidade de seus tipos.

O tipo *char* (constantes *char*) é compatível apenas com ele mesmo. **Seu compilador deve aceitar expressões aritméticas e relacionais com variáveis e literais do tipo *char*. Ou seja, qualquer operação entre operandos *char*, resulta no tipo *char*.**

Os tipos numéricos *float* e *int* são compatíveis, porém não se pode atribuir um *float* a um *int*. Além disso, dividindo-se dois inteiros (variáveis ou constantes) o tipo resultante é *float*

Variáveis devem ter sido declaradas antes de ser usadas, e só podem ser usadas observando-se as regras padrão de escopo. **Não podem haver variáveis com o mesmo nome no mesmo escopo, mas em escopos diferentes (e.g., sub-blocos) são permitidas.**

A tabela de símbolos deve ser utilizada para pesquisa da existência da variável e seu tipo, e deve dar suporte ao mecanismo de escopo explicado no projeto do *parser*.

- Em caso de atribuição ou checagem de variável, se ela não se encontrar no escopo atual, é preciso procurar nos escopos anteriores se ela foi declarada e usar a mais recentemente declarada\*(em caso de múltiplas declarações dela em escopos diferentes). O Erro só ocorre se determinada variável não for achada em nenhum dos escopos.

#### ISSO DEVE FUNCIONAR:

```
{
  int c;
  {
    float y;
    y=y+c;
  }
}
```

#### \*Bônus:

*No caso desse exemplo, no escopo 3, a variável 'c' que será usada na atribuição será o do escopo 2(float c) pois foi declarada mais recentemente se comparado com a do escopo 1(int c).*

```
{
int c; //escopo 1
{
  float c; //escopo 2
  {
    float y; //escopo 3
    y=y+c;
  }
}
}
```

- No caso de declaração de variável, a busca é apenas no escopo atual da declaração, pois não pode ter variáveis iguais no mesmo escopo, gerando erro em caso de múltiplas declarações de uma variável no mesmo escopo.

#### ISSO NÃO DEVE FUNCIONAR:

```
{
int c;
float c;
char c;
}
```

#### ISSO DEVE FUNCIONAR:

```
{
  int c;
  {
    float c;
    {
      char c;
    }
  }
}
```