

## Compiladores – 2016/2

### Roteiro de Laboratório 9

## 1 Objetivos

O objetivo deste laboratório é implementar uma tabela de símbolos e de literais para a linguagem TINY.

## 2 Alterações na Sintaxe da Linguagem TINY

Para melhor exercitar os conceitos deste laboratório, a linguagem TINY deve ser **extendida** com as construções abaixo.

$var-decl \rightarrow \text{int identifier} ;$ $puts-stmt \rightarrow \text{puts string} ;$
--

Uma declaração de variáveis só pode introduzir uma variável por vez e todas são do tipo inteiro. Essas declarações podem acontecer em qualquer ponto do programa mas a semântica da linguagem é alterada para exigir que uma variável seja sempre declarada antes do uso.

O comando `puts` serve para escrever *strings* na tela.

**TAREFA 1:** Modifique o *scanner* e *parser* desenvolvidos no laboratório 7 para incluir os novos elementos da linguagem descritos acima. Baixe do AVA os arquivos de entrada disponibilizados para esse laboratório e teste o seu *parser* modificado.

## 3 Utilizando as Tabelas de Símbolos e de Literais

**TAREFA 2:** Baixe os arquivos de código disponibilizados pelo professor no AVA. Entenda a interface das estruturas descritas no arquivo `.h`. O arquivo `.o` correspondente implementa essas funções. Utilizando o código disponibilizado, inclua no seu *scanner* e *parser* ações para operar com *strings* e variáveis, conforme discutido em sala (veja detalhes nos slides da Aula 06).

O seu analisador semântico deve exibir as seguintes mensagens.

**Utilização de variáveis:** Se uma variável for utilizada sem ser declarada, imprima no terminal:

```
SEMANTIC ERROR (XX): variable 'VV' was not declared.
```

Onde `XX` é o número da linha do programa onde o erro foi detectado e `VV` é o nome da variável.

**Redeclarações de variáveis:** Se uma variável for redeclarada no programa de entrada, imprima no terminal:

```
SEMANTIC ERROR (XX): variable 'VV' already declared at line YY.
```

Onde `XX` é o número da linha do programa onde o erro foi detectado, `VV` é o nome da variável e `YY` é o número da linha aonde a variável foi originalmente declarada.

**Simulando impressão com puts:** Ao reconhecer um comando de `puts` durante a operação de *parse*, imprima no terminal:

PUTS: SS.

Onde SS é a *string* armazenada na tabela de símbolos.

**Impressão das tabelas:** Ao final do *parsing*, imprima as tabelas de símbolos e de literais no terminal. Veja os arquivos de saída disponibilizados no AVA para testes.

**Demais mensagens:** As mensagens de erros léxicos e sintáticos são as mesmas do Laboratório 7 e devem continuar sendo exibidas como antes.

## 4 Implementado as Tabelas de Símbolos e de Literais

**TAREFA 3:** Implemente a sua versão da tabela de símbolos e de literais, substituindo a versão disponibilizada pelo professor. Comece com uma implementação sequencial. Depois, se houver tempo, altere a implementação para uma tabela *hash*.

Observações importantes:

- A princípio, para se realizar as atividades deste laboratório, não é necessário modificar o tipo da variável `yylval`, que pode ficar como inteiro. Nesse caso, talvez seja necessário utilizar uma tabela de símbolos auxiliar para fazer a integração entre o *scanner* e o *parser*.
- Uma outra possibilidade é usar variáveis auxiliares e regras semânticas no meio dos símbolos das regras (veja, por exemplo, o slide 31 da Aula 06).