

Introdução à Compilação

Entrega 3

Bianca Madoka Shimizu Oe
Luis Fernando Dorelli de Abreu
Rafael Umino Nakanishi

1 Objetivo

O objetivo dessa parte final do projeto da disciplina é a implementação do analisador semântico para o compilador para a linguagem LALG disponibilizada para os alunos. A seguir serão discutidos as mudanças que foram necessárias para a implementação do projeto. Nas seções seguintes mostraremos a abordagem utilizada para lidar com a tabela de símbolos e os tratamentos de erros que são gerados pelo compilador. Por fim, estão listados o meio para compilar o código e o método de executá-lo e alguns exemplos que foram utilizados para testar a compilação e suas respectivas saídas.

2 Mudanças realizadas

Sinto muito. Não sei.
Próximo.

3 Decisões de projeto

3.1 Implementação

A implementação do compilador para a linguagem LALG foi feita utilizando o `yacc` (*Yet Another Compiler-Compiler*) em linguagem de programação C. Essa decisão foi influenciada principalmente devido à possibilidade de continuidade a partir da entrega anterior.

((Falar sobre os pseudo-atributos))

3.2 Tabela de símbolos

A tabela de símbolos utilizada nessa implementação é baseado na estrutura de dados *trie*. Nesse caso, cada nó da árvore armazena um caracter correspondente ao nome do identificador. Ao se chegar ao fim da cadeia de caracteres, o nó correspondente armazena os atributos do identificador:

- **type**: tipo do identificador (`integer`, `real`)
- **category**: categoria a que pertence (constante, variável)
- **scope**: escopo do identificador
- **ival**: valor da variável caso seja um número inteiro
- **rval**: valor da variável caso seja um número real
- **paramQty**: quantidade de parâmetros, no caso de procedimentos
- **line**: linha em que foi declarado

- **parameters:** lista de parâmetros, no caso de procedimentos
- **name:** nome completo do identificador, com um máximo de 20 caracteres

3.3 Tratamento de erros

Os seguintes erros são tratados pela análise semântica.

- Variável ou procedimento não declarado
- Variável ou procedimento declarado mais de uma vez
- Incompatibilidade de parâmetros formais e reais: número, ordem e tipo
- Uso de variáveis de escopo inadequado
- Atribuição de um real a um inteiro
- Divisão que entre números não inteiros
- `readln` e `writeln` com variáveis de tipo diferentes

3.4 Geração de pseudocódigo

A geração de código pelo compilador é realizado em paralelo com a análise semântica e é interrompido caso seja encontrado algum erro de natureza léxica, sintática e/ou semântica.

4 Compilação

Para compilação do projeto em sistemas *UNIX*, execute o arquivo `Makefile` disponibilizado por meio do comando `make`. Para o sistema operacional *Windows*, execute a seguinte sequência de comandos:

```
yacc -d -v yacc.yacc --verbose
flex lalg.l
gcc lex.yy.c trie.c symbolTable.c -lfl -o t3.exe
```

Para execução do analisador sintático utilize os comandos:

```
t3.exe <nome_do_arquivo>
```

Observe que o analisador só termina quando encontra um *end-of-file*. Caso o arquivo de entrada esteja gramaticamente correto, obtém-se como saída um arquivo com o *bytecode* referente ao código dado como entrada denominado *code.code*.

5 Exemplos