



Relatório

Informações gerais



DECISÕES

Linguagem compilada: C

Gerador de analisador sintático: Yacc/Lex



DUPLA

Nome: Matheus S. Moreira

Matrícula: 1912947

Nome: Tharik Augusto Lourenço

Matrícula: 1720688

O que foi implementado, como e porquê

Caso 1) Em `varlist` : colocamos `,` como separador e `;` como final da lista:

- Isso foi feito pois o `id` estava guardando mais de um nome.
- Exemplo: `id` vinha `x1 y2` ou `z3;\nSAIDA` em vez desses nomes separados um-a-um

Caso 2) Tiramos os `()` dos cmds `INC` e `ZERA`

- `)` estava vindo colado no nome da variável
- Conseguimos resolver esse problema para o caso da `,` e `;` no nomes das variáveis, mas não nos ocupamos por consertar isso no caso do parênteses foi já estávamos perto do prazo da entrega e ainda tínhamos que nos ocupar com outras avaliações

Caso 3) SAIDA definida como `printf()`

- Apenas uma decisão de projeto

Caso 4) O comando `varlist` foi dividido em `varlist_in` e `varlist_out` :

- Fizemos isso por conta das diferenças de tratamento que implementamos para as listas de variáveis que seguem `ENTRADA` e que seguem `SAÍDA` (ex: guardar os nomes das variáveis para printar no fim do programa, acusar em `SAÍDA` variável não declarada em `ENTRADA`)

Outras informações)

- Implementamos uma lista encadeada para guardar os nomes das variáveis, de modo a conseguir saber quais existem e se foram inicializadas ou não

O que funciona e não funciona



Parece que não há nada que deveria ser minimamente implementado que não esteja funcionando

Testes utilizados



As numerações dos casos de aceitação e erro correspondem aos nomes dos arquivos txt usados como entrada

→ Aceitação

- Aceitação 1: uma chamada de cada comando
- Aceitação 2: diversos comandos chamados e 2 ENQUANTOS aninhados
- Aceitação 3: 3 ENQUANTOS aninhados

→ Erros

Esperava token X que não veio

- Nas linhas de abrir sem fechar Programa/ENQUANTO (erros 1 e 2)
- Na linha de um comando aplicado a nenhuma variável (erros 3 a 7)

Variáveis não-declaradas

- Na linha do ENQUANTO (erro 8)
- Nas linhas dos comandos INC, ZERA e atribuição (erros 9 a 11)
- Na linha da SAÍDA (erro 12)

Variáveis não-inicializadas

- Na linha do ENQUANTO (erro 13)
- Na linha do comando de atribuição (erro 14)

Redeclaração de variáveis (warning)

- Na linha da ENTRADA (erro 15)

Maneira de executar

Compilação

```
yacc -d Sintatico.y; lex Lexico.l; gcc -c lex.yy.c y.tab.c lista.c; gcc -o parser lex.yy.o y.tab.o lista.o -ll
```

Escolha do arquivo de entrada (relação de nomes no tópico de testes utilizados)

```
./parser nomedoarquivo.txt
```


Saída em caso de aceitação

!! O resultado do programa que foi aceito está contido no arquivo saida.txt

Fontes e referências


Token Values (Bison 3.8.1)

4.3.4 Semantic Values of Tokens In an ordinary (nonreentrant) parser, the semantic value of the token must be stored into the global variable `yylval`. When you are using just one data type for semantic values, `yylval` has that type. Thus, if the type is `int` (the default), you might write this in `yylex`: ...

 https://www.gnu.org/software/bison/manual/html_node/Token-Values.html


Union Decl (Bison 3.8.1)

3.4.4 The Union Declaration The `%union` declaration specifies the entire collection of possible data types for semantic values. The keyword `%union` is followed by braced code containing the same thing that goes inside a union in C. For example: `%union { double val; symrec *tpr; }` This says that the two alternative types are `double` and `symrec *`.

 https://www.gnu.org/software/bison/manual/html_node/Union-Decl.html

Type Decl (Bison 3.8.1)

3.7.4 Nonterminal Symbols When you use `%union` to specify multiple value types, you must declare the value type of each nonterminal symbol for which values are used. This is done with a `%type` declaration, like this: Here `nonterminal` is the name of a nonterminal symbol, and `type` is the name given in the `%union` to the alternative that you want (see The Union Declaration).

 https://www.gnu.org/software/bison/manual/html_node/Type-Decl.html