

UNIVERSIDADE FEDERAL DE RORAIMA
DISCIPLINA CONSTRUÇÃO DE COMPILADORES
PROF.: DR. LUCIANO FERREIRA
ALUNO: FELIPE DERKIAN DE SOUSA FREITAS

LISTA 5

BOA VISTA, 14 DE NOVEMBRO DE 2020

5.3 Exercícios

5.1 Um programador C++ codificou o seguinte módulo como parte de sua aplicação:

```
int a;
int umaFuncao() { return a; }
void umaFuncao(int v) { a = v; }
int umaFuncao(int v, int b) { a = v + b; return a; }
```

Após compilar este código com o compilador g++, ele analisou o código objeto associado com o aplicativo nm do sistema Unix, que apresenta a tabela de símbolos do módulo objeto, e encontrou as seguintes definições para os nomes das funções:

```
0000000a T _Z9umaFuncaoI
00000018 T _Z9umaFuncaoIi
00000000 T _Z9umaFuncaoIv
```

- (a) Explique, a partir desse exemplo, como esse compilador faz a decoração de nomes de funções.
- (b) Antes de chegar ao programa anterior, o programador havia tentado compilar, sem sucesso, a seguinte versão de seu código, na qual também utilizava a sobrecarga de nomes de funções:

```
int a;
int umaFuncao() { return a; }
void umaFuncao(int v) { a = v; }
    int umaFuncao(int v) { a = v; return a; }
```

Explique qual foi a causa do problema que o programador encontrou nessa compilação.

5.2 Um programa em C++ contém as seguintes declarações de variáveis:

```
char c = '0';
int i = 1;
float f = 2.0;
```

Quais dos seguintes comandos causariam erros detectados na etapa de análise semântica e, nesse caso, qual tipo de verificação captura o erro?

- (a) `c = i;`
- (b) `i = c;`
- (c) `c % i;`
- (d) `c « i;`
- (e) `f « i;`
- (f) `f % c;`

Questão 5.1:

a) O compilador em questão faz a declaração dos nomes das funções usando a referência da sobrecarga, ou seja, coloca-se um prefixo "_Z9" seguido com o nome da função que nesse caso do exemplo "umaFuncao" e adiciona um posfixo para identificar cada uma das funções pois os nomes são iguais por conta da sobrecarga dos métodos.

b) O que aconteceu é que o programador realizou a sobre carga de funções e utilizou diferentes tipos de retorno sendo void e int o que não é permitido na linguagem.

Questão 5.2:

Causariam erro as alternativas

- **(e) `f << i;`** → Causaria o erro que seria operação com tipos de dados diferentes sendo inteiro e float.
- **(f) `f % c;`** → Causaria o erro que seria operação com tipos de dados diferentes sendo float e char em uma operação de resto da divisão.