Universidade de Évora

Compiladores

Exame de Época Normal - 15 de Junho de 2015

1. Considere uma extensão à linguagem *Ya!*: classes e métodos. Discuta as alterações que terá de fazer relativamente:

(1,5) (a) À análise lexical

(1,5) (b) À análise sintáctica

(1,5) (c) À análise semântica

2. Considere o seguinte programa em *Ya!*:

```
1
2  isPalindrome (s: string, len: int)
3  {
4    if (len < 2) then {
5      return true;
6    }
7    else {
8      return s[0] == s[len-1] and isPalindrome(substring(s, 1, len-1), len-2);
9    };
10 };
11
12 main () : void {
13   print(isPalindrome(``abracarba''));
14 };
15
16
```

(2) (a) Mostre uma representação da Symbol Table, no final da análise semântica da função `isPalindrome()`.

(3) (b) Mostre uma representação da stack, durante a execução do programa, imediatamente antes do return da última chamada (recursiva) da função `isPalindrome()` (quando len==1). Coloque anotações, para especificar o que cada célula da stack representa.

(3) (c) Proponha uma representação intermédia para a função `isPalindrome()`.

(1) (d) Quais são os blocos básicos presentes no código da alínea anterior? Quantos traços diferentes se podem gerar com esses blocos?

3. Aplicando as regras de reescrita para árvores canónicas, reescreva as seguintes árvores:

(1,5) (a) `CALL(NAME(f), [CALL(NAME(g), [e1]), CALL(NAME(g), [e2]), ESEQ(EXP(CALL(NAME(h), [e3])), e4)])`

(1,5) (b) `MOVE(TEMP(t), CALL(NAME(f), [CALL(NAME(f), [ESEQ(s1, e1)]), ESEQ(SEQ(s2, s3), e2), e3]))`

(1,5) (c) `MOVE(e1, MEM(ESEQ(SEQ(s1,s2), e2)))`

(2) 4. Considere a geração de código em formato "máquina de pilha", para a linguagem *Ya!*, estudada nas aulas. Que código gerará a seguinte expressão? (especifique todas as assumpções que tiver de tomar)

```
1
2      c = b = f(a)
3
```