

Paradigmas de Linguagens de Programação
Exame Escrito
Centro de Informática – UFPE, 25 de julho de 2014

Questão 1 [2,0]

- a) Considere uma lista de pares da forma $[(Ana, 2126-8000), (Jose, 3034-5060), (Maria, 9942-2430), \dots]$, representando uma agenda telefônica. Defina uma função **getFone1** que recebe um nome e uma lista (representando uma agenda) e retorna o telefone associado ao respectivo nome. Por exemplo, seja **NF** a lista acima, então **getFone1(Jose, NF) = 3034-5060**. Caso o nome não seja encontrado, a função deve retornar 0 (zero).
- b) Considere uma representação alternativa da agenda telefônica, representada como duas listas de mesmo tamanho, uma de nomes (por exemplo, $[Ana, Jose, Maria, \dots]$) e outra de telefones (por exemplo, $[2126-8000, 3034-5060, 9942-2430, \dots]$). Nesta representação, a associação do telefone a um nome é posicional, ou seja, **Ana** está na posição 1, então seu telefone é **2126-8000**, e assim por diante. Defina uma função **getFone2** que recebe um nome, uma lista de nomes e uma lista de telefones e retorna o telefone associado ao respectivo nome. Por exemplo, seja **N** a lista de nomes e **F** a de telefones acima, então **getFone1(Jose, N, F) = 3034-5060**. Caso o nome não seja encontrado, a função deve retornar 0 (zero).

Questão 2 [2,0] Prove, por indução: **getFone1(n, NF) = getFone2(n, N, F)**

Questão 3 [2,0] Considere o seguinte programa na LI2:

```
{ var y = 1;
  { proc P (par int x) {y := y + 1; x := x + y};
    call P(y);
    write(y)
  }
}
```

Informe o resultado produzido para cada um dos casos quando **par** é interpretado com os seguintes tipos de passagem de parâmetro: a) valor, b) resultado, c) valor-resultado e d) referência.

Questão 4 [1,0] Assinale com V (para verdadeiro) ou F (para falso):

- () É mais natural estender uma linguagem funcional pura com classes e atributos do que com tipos abstratos de dados, considerando o objetivo de manter a transparência referencial.
- () Enquanto variáveis locais e globais de um programa são criadas por declarações, variáveis da **heap** são alocadas dinamicamente e são criadas por comandos.

Questão 5 [3,0] Modifique a linguagem LI1 de forma que a declaração de variáveis inclua o tipo e não envolva uma inicialização, como em **var int x**. Estas variáveis devem ser incluídas no ambiente com um valor especial **nil**, indicando que ainda não foram inicializadas. Se uma variável for usada, em qualquer contexto, antes de ser devidamente inicializada, o programa deve ser interrompido e levantar uma exceção. A implementação deve considerar os métodos de avaliação e checa tipo, bem como as classes novas ou modificadas, impactadas por esta mudança. Particularmente:

- 1) Defina a BNF para a linguagem redefinida, destacando apenas o que mudar.
- 2) Explique se é necessária alguma mudança nos ambientes de compilação e execução.
- 3) Implemente novas classes, se for o caso, e indique todas as classes que seriam afetadas (ilustre a modificação em pelo menos uma classe impactada).

Boa Sorte

Apêndice 1. BNF de LI1.

Programa ::= Comando

Comando ::= Atribuicao

| ComandoDeclaracao

| While

| IfThenElse

| IO

| Comando ";" Comando

| Skip

Skip ::=

Atribuicao ::= Id ":" "=" Expressao

Expressao ::= Valor | ExpUnaria | ExpBinaria | Id

Valor ::= ValorConcreto

ValorConcreto ::= ValorInteiro | ValorBooleano | ValorString

ExpUnaria ::= "-" Expressao | "not" Expressao | "length" Expressao

ExpBinaria ::= Expressao "+" Expressao

| Expressao "-" Expressao

| Expressao "and" Expressao

| Expressao "or" Expressao

| Expressao "==" Expressao

| Expressao "++" Expressao

ComandoDeclaracao ::= "{" Declaracao ";" Comando "}"

Declaracao ::= DeclaracaoVariavel | DeclaracaoComposta

DeclaracaoVariavel ::= "var" Id "=" Expressao

DeclaracaoComposta ::= Declaracao "," Declaracao

While ::= "while" Expressao "do" Comando

IfThenElse ::= "if" Expressao "then" Comando "else" Comando

IO ::= "write" "(" Expressao ")" | "read" "(" Id ")"