Paradigmas de Linguagens de Programação
Exame Escrito
Centro de Informática – UFPE, 13 de junho de 2017

Questão 1 [2,0] Defina uma função imagem relacional, **RI**, que recebe uma lista L1 de pares (de tipos A e B, respectivamente) e uma lista L2 de elementos do tipo A. A função deve retornar uma lista L3 de elementos do tipo B, de forma que, se um par **(a,b)** pertence à primeira lista L1 e **a** pertence a L2, então **b** deve pertencer a L3, com a ordem sendo preservada. Por exemplo: **RI([(a1,b1),(a2,b2),(a3,b3),(a4,b4),(a3,b5)], [a1,a3,a5]) = [b1,b3,b5]**.
[Dica: Definir uma função auxiliar que verifica se um elemento ocorre em uma lista, como na questão de prova resolvida em sala]

Questão 2 [2,0] Lembrando a definição de concatenação

[] ++ ys = ys x:xs ++ ys = x:(xs ++ ys)

prove, por indução, a seguinte propriedade:

RI(xs++ys, zs) = RI(xs, zs) ++ RI(ys,zs)

[Dicas: indução em xs; na prova do passo indutivo, usar análise de casos]

Questão 3 [1,0] Escolha um dos princípios de regularidade no projeto de linguagens, entre os vários apresentados no livro texto, e explique o seu significado e importância.

Questão 4 [2,0] Assinale com V (para verdadeiro) ou F (para falso):

( ) É possível implementar passagem de parâmetro por referência em uma linguagem funcional, sem a perda da transparência referencial.

( ) Considere a linguagem LI1 sem o comando ***while***, mas acrescida de um comando de desvio de fluxo como ***goto L***, onde ***L*** é um label que pode ser associado a um comando para indicar o desvio do fluxo de controle. Neste cenário, LI1 com estas modificações continua sendo considerada uma linguagem de programação.

Questão 5 [3,0] Acrescente à Linguagem Imperativa 2 (veja BNF em anexo), passagem de parâmetro por nome, além da passagem por valor já oferecida pela linguagem. A declaração de um parâmetro por nome deve ser da forma "***name" Tipo Id***. Por exemplo, o procedimento

proc soma (int x, int y, name int z) {z := x + y}

possui parâmetros **x** e **y** por valor e **z** por nome. Na chamada de um procedimento, deve ser sempre passado um identificador de uma variável como parâmetro real, no caso da passagem por nome. Uma possível chamada ao procedimento **soma** seria **some(2,3,w)**, onde **w** deve ser uma variável do tipo **int**. O efeito desta chamada é **w := 2 + 3**. 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. Particularmente, apresente a implementação ou um pseudo código da classe que implementa uma chamada de método neste contexto. Pode assumir a existência de um operador de substituição de uma variável por outra em um comando e que este operador já

resolve conflitos com nomes de variáveis locais. Por exemplo, **c[x/y]** é o comando obtido a partir de **c** substituindo as ocorrências de **y** por **x**.

Boa Sorte

Apêndice 1. BNF de LI2.

Programa ::= Comando

Comando ::= Atribuicao | ComandoDeclaracao
| While | IfThenElse
| IO | Comando ";" Comando
| Skip | ChamadaProcedimento
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 | DeclaracaoProcedimento
| DeclaracaoComposta
DeclaracaoVariavel ::= "var" Id "=" Expressao
DeclaracaoComposta ::= Declaracao "," Declaracao
DeclaracaoProcedimento ::= "proc" Id "(" [ListaDeclaracaoParametro] ")" "{" Comando "}"
ListaDeclaracaoParametro ::= Tipo Id | Tipo Id "," ListaDeclaracaoParametro
Tipo ::= "string" | "int" | "boolean"
While ::= "while" Expressao "do" Comando
IfThenElse ::= "if" Expressao "then" Comando "else" Comando
IO ::= "write" "(" Expressao ")" | "read" "(" Id ")"
ChamadaProcedimento ::= "call" Id "(" [ListaExpressao] ")"
ListaExpressao ::= Expressao | Expressao, ListaExpressao

Classes Auxiliares

AmbienteCompilacaoImperativa2

AmbienteExecucaoImperativa2
ContextoCompilacaoImperativa2
ContextoExecucaoImperativa2
ListaValor
Procedimento
DefProcedimento
ProcedimentoJaDeclaradoException
ProcedimentoNaoDeclaradoException