

Paradigmas de Linguagens de Programação

Exame Escrito

Centro de Informática - UFPE

22 de junho de 2010

Questão 1 [2,0] Defina uma função *merge* que recebe duas listas ordenadas de inteiros e retorna uma lista ordenada com todos os elementos das listas recebidas como parâmetro. Por exemplo, *merge*([5,7,9],[3,4,6,8,9]) = [3,4,5,6,7,8,9,9]

Questão 2 [2,0] a) Defina uma propriedade da função *merge* que garante que o tamanho da lista resultante é a soma dos tamanhos das listas de entrada
b) prove esta propriedade para o caso base.

Questão 3 [2,0] Explique como as linguagens de programação evoluíram a partir de mecanismos de abstração cada vez mais elaborados. Dê exemplos de mecanismos de abstração em linguagens modernas e explique o que sugere o *princípio da abstração* como apresentado no livro texto do curso.

Questão 4 [4,0] Acrescente o conceito de tupla na Linguagem Imperativa 1, considerando que uma tupla é um par de elementos, possivelmente heterogêneos, como *(1,true)*, *("PLP",0)*, *(3,4)*, ou mesmo tupla de tuplas, como em *((1,true),"abc")*. Defina operadores que retornam o primeiro e o segundo elementos de uma tupla:

pri(a,b) = a seg(a,b) = b

- Defina a BNF para a linguagem redefinida, destacando apenas o que mudar.
- Explique o que precisaria mudar no *parser* e se é necessária alguma mudança nos ambientes de compilação e execução.
- Implemente as novas classes que se mostrem necessárias e modifique as existentes quando necessário.

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 ")"