Universidad del Valle

Solución Primer examen parcial - Fundamentos de lenguajes de programación

Duración: 2 horas

Carlos Andres Delgado S, Ing *

24 de Abril 2017

Nombre:_____
Código:_____

1. Especificación recursiva de datos [20 puntos]

1. (10 puntos) Escriba la especificación inductiva de las listas de los números múltiplos de 7.
   Dado S como el conjunto de los múltiplos de 7 y M un número múltiplo de 7

$$\frac{0 \in M, '() \in S}{n \in M \wedge l \in S}$$
$$(cons(n+7), l)$$

2. (10 puntos) Escriba la especificación gramatical de las listas de parejas de números. Tenga en cuenta que **empty** es la lista vacía.

```
<lista-parejas> := {parejas}*

<pareja>:= {numero numero}*
```

2. Abstracción de datos [50 puntos]

Dada la siguiente gramática:

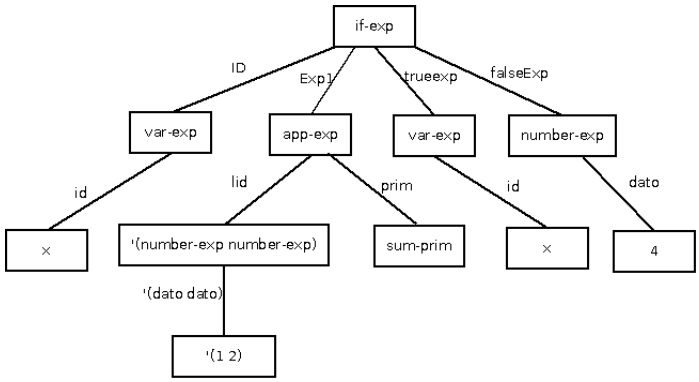
```
expresion:= identificador
            var-exp (id)
         := numero
            number-exp (dato)
         := "execute" "("expresion* (',')")" "do" primitiva "end"
            prim-exp (lid prim)
         := "if" "("identificador "=" expresion ")" "then" expresión "else" expresion
            if-exp (id exp1 trueExp falseExp)

identificador := simbolo+

numero := digito+

primitiva := "+"
           sum-prim
           := "-"
           minus-prim
```

En este caso los valores denotados son **símbolos**

1. (10 puntos) Construya el árbol de sintaxis abstracta para la siguiente expresión:

```
if ( x = execute (1,2) do + end ) then x else 4
```

*carlos.andres.delgado@correounivalle.edu.co



2. (15 puntos) Utilizando **define-datatypes** defina la gramática de la expresión.

```
(define-datatype expresion expresion?
  (var-exp (id symbol?))
  (number-exp (dato number?))
  (prim-exp (lid (list-of expresion?))
            (prim primitiva?))
  (if-exp (id symbol?)
          (exp1 expresion?)
          (trueExp expresion?)
          (falseExp expresion?))
  )

(define-datatype primitiva primitiva?
  (sum-prim)
  (minus-prim)
  )
```

Para el caso de identificador y número utilice **symbol?** y **number?** respectivamente.

3. (5 puntos) Utilizando la definición del **datatype** de expresión construya una expresión abstracta de **prim-exp**

```
(prim-exp (list (number-exp 3) (number-exp 9)) (sum-prim))
```

4. (20 puntos) Genere una función **evaluar-expresion** la cual recibe una expresión y un ambiente. Tenga en cuenta:

   - Los valores expresados son **números**.
   - En el caso del if **identificador = expresion**, ya existe **evaluar-condicional** que los recibe y retorna falso o verdadero.
   - Ya existe un tipo de dato para ambiente
   - Ya existe una función **apply-env**

```
(define evaluar-expresion
    (lambda (exp amb)
        (cases expresion exp
            (var-exp (id) (apply-env id amb))
            (number-exp (dato) dato)
            (prim-exp (lid prim) (aplicar-primitiva (lid prim)))
            (if-exp (id exp1 trueExp falseExp)
                  (if (evaluar-condicional id (evaluar-expresion exp1))
                          (evaluar-expresion trueExp)
                          (evaluar-expresion falseExp)
                    )
             )
          )
      )
  )

 (define aplicar-primitiva
        (lambda (lexp prim)
           (cases primitiva prim
               (sum-prim () (realizar-operacion lexp +))
               (minus-prim () (realizar-operacion lexp -))
            )
         )
 )
```

```
(define realizar-operacion
  (lambda (lista op)
    (cond
      [(null? lista) 0]
      [else (op (evaluar-expresion (car lista)) (realizar-operacion (cdr lista) op))]
      )
    )
  )
```

3. Evaluación de expresiones [30 puntos]

Considere la siguiente expresión en el lenguaje visto en el curso (procedimientos), con ambiente inicial *env0* con identificadores **(a b c g)** y valores **(2 3 4 (closure'(x y z) +(x,*(y z)) empty-env))**

```
let
   a = (g a b c)
   b = (g b c a)
   c = (g c a b)
   in
      let
            f = proc(a,b) if a then b else *(a,b)
            in
              let
                 a = (f a b)
                 b = (f a c)
                 in
                    +(a,b,c)
```

1. (5 puntos) Indique el valor de la expresión.
2. (25 puntos) Dibuje los ambientes que se generan y muestre mediante líneas de que ambientes extienden.