Control 1 – Lenguajes de Programación (CC41A)

Departamento de Ciencias de la Computación – Universidad de Chile

Profesor: Éric Tanter

9 de Abril 2007

Evaluación: 1 punto por pregunta

1. Implemente en Scheme la función `revrel` que invierte una relación dada (una relación es una lista de pares):

```
(revrel '((1 2)))             --> ((2 1))
(revrel '((1 2) (3 4) (5 6))) --> ((2 1) (4 3) (6 5))
```

2. Considere la siguiente implementación de la función de substitución `subst` del lenguaje WAE visto en clases:

```
;; subst: WAE symbol WAE -> WAE
(define (subst expr sub-id val)
  (type-case WAE expr
    (num (n) expr)
    (add (l r) (add (subst l sub-id val)
                    (subst r sub-id val)))
    (sub (l r) (sub (subst l sub-id val)
                    (subst r sub-id val)))
    (with (bound-id named-expr bound-body)
          (if (symbol=? bound-id sub-id)
              expr
              (with bound-id
                    named-expr
                    (subst bound-body sub-id val))))
    (id (v) (if (symbol=? v sub-id) val expr))))
```

   Con esta función de substitución, el calculador se cae en las dos expresiones siguientes:

   a) `{with {x 5} {with {y x} y}}`

   b) `{with {x 5} {with {x x} x}}`

   Para cada caso explique porqué se cae el calculador, y luego de una versión corregida de `subst`.

3. ¿Que régimen de substitución implementa el siguiente calculador? ¿Que habría que cambiar si uno quisiera pasar al otro régimen?

```
;; calc : WAE -> number
(define (calc expr)
  (type-case WAE expr
     (num (n) n)
     (add (l r) (+ (calc l) (calc r)))
     (sub (l r) (- (calc l) (calc r)))
     (with (bound-id named-expr bound-body)
           (calc (subst bound-body
                        bound-id
                        named-expr)))
     (id (v) (error 'calc "free identifier"))))
```

4. ¿Puede probar que el régimen de substitución en un lenguaje como WAE no afecta el resultado de la evaluación de un programa? ¿Para que lenguajes podría tener un efecto, tal que cambiar de régimen de substitución cambie el valor de una expresión? De un ejemplo.

5. Considere un lenguaje donde la ejecución del siguiente programa retorna el valor 5:

```
(define (f p) n)
(local ((define n 5))
  (f 10))              --> retorna 5
```

   ¿Como caracterizaría a tal lenguaje? ¿En que difiere de lo que esperaría un programador Java? ¿Que modificación haría a su interprete para volver a la semantica esperada?

6. En un intento de implementar el interprete del lenguaje F1WAE, tenemos errores con ciertas expresiones con `with`. Aquí va su interprete:

```
;; interp : F1WAE listof(fundef) DefrdSub -> number
(define (interp expr fun-defs ds)
  (type-case F1WAE expr
    (num ...) (add ...)
    (with (bound-id named-expr bound-body)
          (interp bound-body fun-defs
                  (aSub bound-id
                        (interp named-expr fun-defs ds)
                        (mtSub))))
    (id ...) (app ...)))
```

   De *a)* un ejemplo de expresión con `with` que sí funciona, *b)* un ejemplo de expresión con `with` que no funciona, *c)* la implementación corregida de la interpretación del `with`.