

# Aux C3 – Lenguajes de Programación (CC41A)

Departamento de Ciencias de la Computación – Universidad de Chile

Profesor: Éric Tanter  
Auxiliar: Oscar E. A. Callaú

5 de Noviembre del 2008

## 1. C3, Otoño 07

1. (1pt) Ud. recién descubrió un nuevo lenguaje cuyas características desconoce. ¿Qué puede decir del resultado del siguiente programa (la función `main` es el punto de entrada del programa)? Explique.

```
def y = 10;
def void foo(x) { x = y; }
def void main() {
  def z = 5;
  foo(z);
  println(z);
}
```

2. (1.5pt) Considere el siguiente programa:

```
{with {f {fun {x}
          {fun {y} {+ x y}}}}
  {+ 3 {f 5}}}
```

- a) Reescriba este programa con las anotaciones de tipos necesarias.
- b) Escriba y explique el juicio de tipo de una expresión `with`.
- c) Demuestre, desarrollando el árbol de juicios de tipos, que el programa anterior **no** es válido.

## 2. C3, Primavera 07

Suponga que Ud. desea ejecutar un programa estrictamente funcional. Discuta cómo podría ser afectado el resultado del programa por el paso de parámetros a funciones en los siguientes lenguajes:

1. Scheme
2. Un dialecto de Scheme que pasa los parámetros por valor
3. Un dialecto de Scheme que pasa los parámetros por referencia

### 3. C3, Otoño 08

1. Considere el siguiente programa:

```
{with {fact {fun {n} {if0 n 1 {* n {fact {- n 1}}}}}
  {fact 10}}
```

- a) De el equivalente de este programa *sin with*, usando la equivalencia entre **with** y otras expresiones del lenguaje.
- b) Describa su evaluación en un interprete con *substitución explícita* (es decir, usando **subst**).
- c) Describa su evaluación en un interprete con *substitución diferida* (es decir, usando un ambiente).

2. Tipos

- a) En base al juicio de tipos del **with** proponga un juicio apropiado para el *with multiple*:

```
<expr> ::= ...
         | {with {{t <id> <expr>}*} <body>}}
```

Ejemplos de expresiones son:

```
{with {{int a 1} {int b 1}}
  {+ a b}}
```

```
{with {{int->int f {fun {int x} : int {+ x x}} {int c 1}}
  {f {f {f c}}}}
```

- b) Ahora que conoce como usar el *with multiple* determine mediante un árbol de derivación si la siguiente expresión es válida:

```
{with {{int->int f {fun {int x} : int {x}}}
  {int y 1}}
  {{f f} y}}
```

- c) Re-escriba la expresión anterior sin notación de tipo y explique (justificando) ¿que sucede si esa expresión se evalua con interprete de un lenguaje de tipado dinámico? ¿con un lenguaje provisto de un sistema de inferencia de tipos (como Haskell y ML)?