

Lenguajes de Programación (CC41A) - Primavera 2009

Clase Auxiliar 9

Profesor: Tomás Barros

Auxiliar: Víctor Ramiro

1. Considere el siguiente lenguaje (similar al FAE visto en clases):

```
<expr> ::= <id>
          | <num>
          | (+ <expr> <expr>)
          | (lambda (<id>) <body>)
          | (<expr> <expr>)
```

- Exprese, en este lenguaje, un programa que nunca termina.
- Especifique los juicios de tipos para las distintas expresiones de este lenguaje. No es necesario asumir que los programas llevan anotaciones de tipo (ie. puede suponer que tiene un inferenciador de tipos).
- ¿Es posible asignar un tipo a su programa que nunca termina? En caso en que no, esto significa que tiene un lenguaje (FAE+tipos) que asegura que todos los programas validos en tipos siempre terminan. ¿No es eso contradictorio con el hecho de que los sistemas de tipos son sujetos al Halting Problem? Comente.

2. Considere el siguiente programa:

```
{with {f {fun {x : num} : (num -> num)
          {fun {y : num} : num
            {+ x y}}}}
  {+ 3
    {f 5}}}
```

- a) Explique las anotaciones de tipo del programa.
 - b) Escriba y explique los juicios de tipo de las siguientes expresiones: **with**, **fun** (definición), y **app**.
 - c) Demuestre, desarrollando los juicios de tipos, que el programa anterior **no** es válido.
3. 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:
 - a) Scheme
 - b) Un dialecto de Scheme que pasa los parámetros por valor
 - c) Un dialecto de Scheme que pasa los parámetros por referencia

4. Escriba el resultado esperado para el siguiente programa, justificando en cada paso los valores en el Environment y en el Store.

```
{with {setter {refun {x}
                  {set x 41}}}}
  {with {y 10}
    {with a {newbox y}}
      {seqn
        {setter y}
        {+ {openbox a} y}}}}}
```