

Universidad Nacional Autónoma de México
Facultad de Ciencias
Lenguajes de Programación 2017-1
Tarea 5

Karla Ramírez Pulido
José Ricardo Rodríguez Abreu
Manuel Soto Romero

Fecha de inicio: 8 de Noviembre.
Fecha de entrega: 16 de Noviembre.

1 Lineamientos de la tarea

1. Esta tarea puede ser entregada en equipos de máximo 3 integrantes.
2. Se recibirá a mano con letra legible o de manera digital escrito en \LaTeX en cuyo caso se deberá enviar el archivo `.tex` y `.pdf` al correo `ricardo_rodab@ciencias.unam.mx` con el asunto `[lenguajes]Tarea05`.
3. En caso de que alguna respuesta involucre escribir algún pequeño programa, este será enviado a más tardar el día de la fecha de entrega al correo `ricardo_rodab@ciencias.unam.mx` con el asunto `[lenguajes]Tarea05-Codigos` y cada archivo será nombrado con el número de pregunta que le corresponda.
4. La fecha de entrega es única y no habrá prórroga.
5. Se deberá anexar al final de la tarea la bibliografía consultada. Cualquier tarea con material consultado y no citado propiamente se le considerará plagio y será promediado con cero.

2 Responda correctamente cada inciso

- 2.1** (1 ptos) Hacer el juicio de tipo para la función `fibonacci` y el predicado `empty?`.
- 2.2** (1 pto) Considerar el siguiente programa:

```
(+ 1 (first (cons true empty)))
```

Este programa tiene un error de tipos. Genera restricciones para este programa. Aísla el conjunto mas pequeño de estas restricciones tal que, juntas, identifiquen el error de tipos. Siéntete libre de etiquetar las sub-expresiones del programa con super índices para usarlos cuando escribas y resuelvas tus restricciones.

2.3 (1 pto) Escribir el juicio de tipos para las siguientes expresiones:

a) `let*` del lenguaje Racket.

b) `with` del lenguaje FWAE.

2.4 (2 ptos) Con base a la función `interp` de la página 131, vista en clase, interpretar la siguiente función usando Store-Passing Style:

```
{with {x {newbox 6}}
      {{fun {y} {if0
                {seqn {setbox x 2}
                      {seqn {with {n {newbox 6}} 10}
                            {+ -2 {openbox x}}}}}
        {+ y {openbox x}}
        {+ {openbox n} {openbox x}}}} 1}}
```

2.5 (1 pto) Responder correctamente con base al siguiente texto:

Considera los juicios de tipos discutidos en clase para un lenguaje glotón (en el capítulo de Juicios de Tipos del libro de Shriram). Considera ahora la versión perezosa del lenguaje. Pon especial atención a las reglas de tipado para:

- Definición de funciones
- Aplicación de funciones

Para cada una de estas, si crees que la regla original no cambia, explica por que no (Si crees que ninguna de las dos cambia, puedes responder las dos partes juntas). Si crees que algún otro juicio de tipos debe cambiar, menciónalo también.

2.6 (2 ptos) Desarrolla la inferencia del tipos de la siguiente expresión en el lenguaje RCFAE:

```
{rec {fib {fun {n}
           {if {<= n 2}
               1
               {+ {fib {- n 1}} {fib {- n 2}}}}}
      {fib 10}}
```

2.7 (1 pto) ¿Cuáles son las ventajas y desventajas de tener polimorfismo explícito e implícito en los lenguajes de programación? (Mencionar dos ventajas y dos desventajas)

2.8 (1 pto) Considera la siguiente expresión con tipos:

```
{fun {f : C1 } : C2
  {fun {x : C3 } : C4
    {fun {y : C5 } : C6
      {cons x {f {f y}}}}}}
```

Dejamos los tipos sin especificar (C_n) para que sean llenados por el proceso de inferencia de tipos. Deriva restricciones de tipos para el programa anterior. Luego resuelve estas restricciones. A partir

de estas soluciones, rellena los valores de las Cn. Asegúrate de mostrar todos los pasos especificados por los algoritmos (i.e., escribir la respuesta basándose en la intuición o el conocimiento es insuficiente). Deberás usar variables de tipo cuando sea necesario. Para no escribir tanto, puedes etiquetar cada expresión con una variable de tipos apropiada, y presentar el resto del algoritmo en términos solamente de estas variables de tipos.

2.9 (1 pto) Utilizar el algoritmo de Unificación visto en clase para las siguientes expresiones:

Expresión 1:

```
( (lambda(x)
  (* x (+ x x)))
  (/ 10 (* 2 5)))
```

Expresión 2:

```
( (lambda(y) (/ y y))
  (- 10 (* 2 5)))
```

3 Formato bibliografía

El formato que debe llevar la bibliografía es el siguiente:

Nombre del autor empezando por apellidos, "Nombre del libro", Editorial, Edición, País, No. de páginas del libro.

ó

Nombre del sitio oficial y posible autor (si es que lo hay). URL. Consultado el día:[Fecha de consulta]