

Tarea 1

Profesora: Karla Ramírez Pulido

Ayudante: Héctor Enrique Gómez Morales

Fecha de inicio: 8 de septiembre de 2015

Fecha de entrega: 18 de septiembre de 2015

1. Problema I

Hemos visto en clase que la definición de sustitución resulta en una operación ineficiente: en el peor caso es de orden cuadrático en relación al tamaño del programa (considerando el tamaño del programa como el número de nodos en el árbol de sintaxis abstracta). También se vio la alternativa de diferir la sustitución por medio de ambientes. Sin embargo, implementar un ambiente usando un stack no parece ser mucho más eficiente.

Responde las siguientes preguntas.

- Provee un esquema para un programa que ilustre la no-linealidad de la implementación de ambientes basada en un stack. Explica brevemente porque su ejecución en tiempo no es lineal con respecto al tamaño de su entrada.
- Describe una estructura de datos para un ambiente que un intérprete de FWAE pueda usar para mejorar su complejidad
- Muestra como usaría el intérprete esta nueva estructura de datos.
- Indica cual es la nueva complejidad del intérprete (análisis del peor caso) y de forma informal pero rigurosa pruébalo.

2. Problema II

Dada la siguiente expresión de FWAE:

```
{with {x 4}
  {with {f {fun {y} {+ x y}}}
    {with {x 5}
      {f 10}}}}
```

debe evaluar a (*num* 14) usando alcance estático, mientras que usando alcance dinámico se obtendría (*num* 15), Ahora Ben un agudo pero excéntrico estudiante dice que podemos seguir usando alcance dinámico mientras tomemos el valor más viejo de *x* en el ambiente en vez del nuevo y para este ejemplo el tiene razón.

- ¿Lo que dice Ben está bien en general? si es el caso justifícalo.
- Si Ben está equivocado entonces da un programa de contraejemplo y explica por que la estrategia de evaluación de Ben podría producir una respuesta incorrecta.

3. Problema III

Dada la siguiente expresión de FWAE con `with` multi-parametrico:

```
{with {{x 5} {adder {fun {x} {fun {y} {+ x y}}}} {z 3}}
      {with {{y 10} {add5 {adder x}}}
            {add5 {with {{x {+ 10 z}} {y {add5 0}}}
                  {+ {+ y x} z}}}}}
```

- Da la forma Bruijn de la expresión anterior.
- Realiza la corrida de esta expresión, es decir escribe explícitamente cada una de las llamadas tanto para `subst` y `interp`, escribiendo además los resultados parciales en sintaxis concreta.