Tarea 1

Profesora: Karla Ramírez Pulido Ayudante: Héctor Enrique Gómez Morales
Fecha de inicio: 8 de septiembre de 2015
Fecha de entrega: 21 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 ambientes. Sin embargo, implementar un ambiente usando un stack no parece ser mucho mas 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.

```
{with {u 4}
        {with {v 5}
                {with {w 6}
                        {with {x 6}
                                {with {y 6}
                                        {with {z 6}
                                                {+ u u u u u u}}}}}
```

Se ejemplifica por qué el stack quedaría
((z 6) (y 6) (x 6) (w 6) (v 5) (u 4))
para lo cual para hacer una búsqueda de su valor con un look-up recorrería la lista tantas veces como se requiera u, donde si el programa tiene como nodos del árbol de sintaxis abstracta tantas variables como (n - 1), lo recorrería 6 veces en efecto sería mucho más grande que O(n).

- Describe una estructura de datos para un ambiente que un intérprete de FWAE pueda usar para mejorar su complejidad

Si usamos como estructura una tabla hash con el id como llave, en el peor de los casos si aparece k veces el id tenemos k colisiones, tendríamos que buscar a lo mas k veces en cada ocasión con número de línea más cercano lo que es de orden lineal. Suponiendo que guardamos (id valor #línea) para saber por cual valor hay que tomar dependiendo si es dinamico o estatico

- Muestra cómo usaría el intérprete esta nueva estructura de datos.

Necesitariamos la función lookup modificarla para pasarle como argumentos el id, numero de linea actual y el ambiente de tabla hash, el cual buscaría según el id, si hay colisión, otra función auxiliar buscaría en la lista de colisiones el id con el #linea más cercano a la línea actual y devolvería ese valor a lookup que a su vez lookup regresaría

- Indica cuál es la nueva complejidad del intérprete (análisis del peor caso) y de forma informal pero rigurosa pruébalo.

En el peor de los casos si aparece k veces el id tenemos k colisiones, tendríamos que buscar a lo mas k veces en cada ocasión con el numero de linea mas cercano que es de orden lineal.

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 esta bien en general? si es el caso justificalo.

Ben esta equivocado

- Si Ben está equivocado entonces da un programa de contraejemplo y explica por qué la estrategia de evaluación de Ben podría producir una respuesta incorrecta.

Si tenemos

```
{with {x 3}
        {with {x 4}
                {with {f {fun {y} {+ x y}}}
                        {with {x 5}
                                {f 10}}}}}
```

La expresión con alcance estático daría (num 14) y con alcance dinámico (num 15),
pero con la suposición de Ben al final el ambiente es (num 13)
env= ((y 10) (x 5) (f {fun {y} {+ x y} } ((x 4) (x 3))) (x 4) (x 3))

la función f busca su valor en el closure(ambiente) por tanto x lo toma como 4, lo cual devuelve (num 14) pero si tomamos la suposición de Ben, f buscaría el valor más viejo en el closure(ambiente) y tomaría x con 3 y eso regresaría (num 13) pero no queremos eso.

3. Problema III
Dada la siguiente expresión de FWAE con with multi-paramétrico:

```
{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.

```
{with 5 {fun {x} {fun {y} {+ x y }}} 3
        {with 10 {adder x}
                { <: 0 1>
                        {with {+ 10 <:3 3>} {<:1 1> 0}
                                {+ {+ <:1 1> <:1 0>} <:3 2>}}}}}}
```

- 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.

Para que el intérprete corra el with multiparamétrico, lo convertí al de FWAE y queda de la siguiente manera:

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