

Lenguajes de Programación

Auxiliar N°4

Auxiliares: Kenji Maillard

14/04/2020

1 Resumen

En estos ejercicios consideramos los dos lenguajes visto en clase:

1. El language F1WAE tiene dos ambientes: un de numeros (valor de primera clase) y un de funciones (de segunda clases).
2. El language FWAE tiene funciones de primera clase: hay solamente un ambiente y valores pueden ser numeros o funciones.

2 Ejercicios

1. *Scope estatico y dinamico*. Considere el siguiente programa en F1WAE:

```
(define my-funcs (list (fundef 'foo 'y (parse 'x))))  
  
(run '(with (x 1)  
      (+ (with (x 2) x)  
        (foo 3))) my-funcs)
```

- ¿Cual es el resultado del programa con scope estatico? Argumente.
 - ¿Cual seria el resultado del programa anterior con scope dinamico?
2. Considere el siguiente programa:

```
(let ([x 0])
  (let ([f (λ () x)])
    (let ([x 42]) (f))))
```

- ¿Cual es el resultado del programa?. Argumente
- Si Racket tuviera Scope dinamico por defecto, ¿Cual seria el resultado del programa anterior?

3. *Eliminación de with.* Con funciones de primera clase en FWAE, se puede definir la construction

```
{with {id expr} body}
```

con

```
{{fun id body} expr}
```

Define una función tomando una expresión y eliminando los `with`.

4. *Función de orden superior y tipo de datos.* En el language FWAE, considera las expresiones

```
(define t (parse '{fun x {fun y x}}))
(define f (parse '{fun x {fun y y}}))
(define e (parse '{fun h {fun x {fun y {{h x} y}}}}))
```

- ¿Cuáles son los resultados de las expresiones `{{{e t} 3} 5}` y `{{{e f} 3} 5}`?
- ¿Qué relación tienen estas funciones con booleanos? Define la función `and` con `t`, `f` y `e`.
- ¿Es posible definir otros tipos de datos (pares, sumas, naturales, arboles,...)? Argumente.

5. *Un poco de λ-calculus.* ¿Es posible definir funciones recursivas en el lenguaje FWAE? Argumente.