

Trabajo Práctico 3

Programación Orientada a Objetos

Paradigmas de Lenguajes de Programación — Verano 2018

Fecha de entrega: 1 de Marzo del 2017

Introducción

En este trabajo vamos a modelar el cálculo lambda simplemente tipado con objetos. Consideraremos el siguiente fragmento:

$$\begin{aligned}\sigma &::= \text{Bool} \mid \sigma \rightarrow \sigma \\ M &::= \text{true} \mid \text{false} \mid MM \mid \lambda x : \sigma. M \mid x\end{aligned}$$

Ejercicios a desarrollar

Ejercicio 1

Proveer los objetos y/o constructores necesarios que permitan generar al conjunto de expresiones de tipos y términos del lenguaje. Las constantes `Bool`, `true`, `false` estarán representadas respectivamente por los objetos `Bool`, `TT` y `FF` (su solución no debe basarse en el tipo `boolean`). Deberán definirse las funciones constructoras adecuadas para generar a los restantes términos del lenguaje.

Ejercicio 2

Extender las definiciones previas para que sea posible obtener una descripción textual de cada término del lenguaje. Por ejemplo, si `m` es un objeto que representa al término $\lambda x : (\text{Bool} \rightarrow \text{Bool}) \rightarrow \text{Bool}.(x \text{ true})$, la expresión `m.toString()` evalúa

```
"\x:((Bool -> Bool)-> Bool).(x true)".
```

Ejercicio 3

Implementar un mecanismo de *deep copy* (para los términos del lenguaje) que permita generar una copia del receptor de manera tal que el mismo y su copia no compartan referencias a los objetos que representan subtérminos. Si `o` representa a un término o a una expresión de tipo, entonces debe valer que `o != o.deepCopy()` y `o.toString() == o.deepCopy().toString()`.

Ejercicio 4

Extender las definiciones precedentes de manera tal que todos los objetos que representan términos del lenguaje puedan responder al mensaje `sust("x",m)`, cuyo efecto es sustituir todas las ocurrencias libres de la variable x por copias frescas del término m . Puede asumir que m es siempre un término cerrado. La sustitución puede modificar al objeto receptor.

Ejercicio 5

Extender las definiciones dadas de manera tal que todos los objetos que no representan valores puedan responder al mensaje `reducir()`. Cuando un objeto recibe el mensaje `reducir()` aplica, si es posible, la regla de reducción sobre el redex indicado por la estrategia de evaluación *call-by-value*. Es importante que un paso de reducción sólo reduzca a un redex. La reducción puede modificar al objeto receptor.

Importante: los términos que denotan valores no deben poder responder a este mensaje.

Ejercicio 6

Extender su solución de manera tal que toda expresión pueda responder al mensaje `evaluar` que reduce al objeto receptor a su forma normal. Su solución debe definir una cadena adecuada de prototipos y utilizar herencia de manera tal que la definición `evaluar()` se realiza en un único objeto.

Ejercicio 7

Redefinir el mecanismo de *deep copy* del ejercicio 3 en términos de un *Trait*. Para poder tratar uniformemente a los distintos objetos, su solución puede iterar sobre los atributos de un objeto (`for (k in o){...}`).

Pautas de entrega

Se debe entregar el código impreso con la implementación de las funciones pedidas. Cada función asociada a los ejercicios debe contar con ejemplos que muestren que exhibe la funcionalidad solicitada. Además, se debe enviar un e-mail conteniendo el código fuente en Javascript a la dirección plp-docentes@dc.uba.ar. Dicho mail debe cumplir con el siguiente formato:

- El título debe ser [PLP;Taller-00] seguido inmediatamente del nombre del grupo.
- El código Haskell debe acompañar el e-mail y lo debe hacer en forma de archivo adjunto (puede adjuntarse un `.zip` o `.tar.gz`).
- El código entregado **debe** incluir tests que permitan probar las funciones definidas.

No es necesario entregar un informe sobre el trabajo, alcanza con que el código esté **adecuadamente** comentado (son comentarios adecuados los que ayudan a entender lo que no es evidente o explican decisiones tomadas; no son adecuadas las traducciones al castellano del código). Los objetivos a evaluar son:

- Corrección.
- Declaratividad.

Importante: se admitirá un único envío, sin excepción alguna y solo se aceptaran grupos de dos personas.