Trabajo Práctico 4
Análisis de Lenguajes de Programación
Licenciatura en Ciencias de la Computación

Morales, Sebastián
Moliné, Sebastián

6 de diciembre de 2020

1. Ejercicio 1a

- **monad.1:** return $x \gg\!= f \;\equiv\; f\ x$

$$\text{return } x \gg\!= f$$

$=\{ \gg\!=,1 \}$

$$\begin{gathered} State\ (\lambda s \to let\ (v :! : s') = runState\ (return\ x)\ s \\ in\ runState\ (f\ v)\ s') \end{gathered}$$

$=\{$ return.1 $\}$

$$\begin{gathered} State\ (\lambda s \to let\ (v :! : s') = \\ runState\ (State\ (\lambda s1 \to (x :! : s1))\ s \\ in\ runState\ (f\ v)\ s')) \end{gathered}$$

$=\{$ runState . State $\equiv$ id $\}$

$$\begin{gathered} State(\lambda s \to let\ (v :! : s') = (\lambda s1 \to (x :! : s1))\ s \\ in\ runState\ (f\ v)\ s')) \end{gathered}$$

$=\{$ β-redex $\}$

$$\begin{gathered} State(\lambda s \to let\ (v :! : s') = (x :! : s) \\ in\ runState\ (f\ v)\ s') \end{gathered}$$

$=\{$ let $\}$

$$State(\lambda s \to runState\ (f\ x)\ s)$$

$=\{$ β-redex $\}$

$$State(runState(f\ x))$$

$=\{$ State . runState $\equiv$ id $\}$

$$f\ x$$

- **monad.2:** $t \gg\!= return \;\equiv\; t$

$$t \gg\!= return$$

$=\{ \gg\!=,1 \}$

$$\begin{gathered} \text{State}\ (\lambda s \to let\ (v :! : s') = runState\ t\ s \\ in\ runState\ (return\ v)\ s') \end{gathered}$$

$=\{$ return.1 $\}$

$$\begin{gathered} State(\lambda s \to let\ (v :! : s') = runState\ t\ s \\ in\ runState\ (State\ (\lambda s1 \to (v :! : s1))\ )\ s') \end{gathered}$$

$=\{$ runState . State $\equiv$ id $\}$

$$State(\lambda s \rightarrow let\ (v :! : s') = runState\ t\ s$$
$$in\ (\lambda s1 \rightarrow (v :! : s1))\ s')$$

={ β-redex **}**

$$State(\lambda s \rightarrow let\ (v :! : s') = runState\ t\ s$$
$$in\ (v :! : s'))$$

={ let **}**

$$State(\lambda s \rightarrow runState\ t\ s)$$

={ β-redex **}**

$$State(runState\ t)$$

={ State . runState $\equiv$ id **}**

$$t$$

- **monad.3:** $t \gg= return\ \equiv\ t$

  Se desarrolló desde ambos lados de la igualdad para concluir que, al cumplirse la misma, la propiedad se cumple.

  Del lado izquierdo de la igualdad se tiene:

$$(t\ \gg= f) \gg= g$$

={ $\gg=$,1 **}**

$$State(\lambda s \rightarrow let\ (v :! : s') = runState(t \gg= f)\ s$$
$$in\ runState(g\ v)\ s')$$

={ $\gg=$,1 **}**

$$State(\lambda s \rightarrow let\ (v :! : s') = runState(State\ (\lambda s_0 \rightarrow$$
$$let\ (v_0 :! : s_0') = runState\ t\ s_0$$
$$in\ runState\ (f\ v_0)\ s_0'))\ s$$
$$in\ runState(g\ v)\ s')$$

={ runState . State $\equiv$ id **}**

$$State(\lambda s \rightarrow let\ (v :! : s') = (\lambda s_0' \rightarrow$$
$$let\ (v :! : s_0') = runState\ t\ s_0'$$
$$in\ runState\ (f\ v_0)\ s_0'))\ s$$
$$in\ runState\ (g\ v)\ s')$$

={ β-redex **}**

$$State(\lambda s \rightarrow let(v :! : s') =$$
$$let\ (v :! : s_0') = runState\ t\ s$$
$$in\ runState(f\ v_0)\ s_0')$$
$$in\ runState(g\ v)\ s')$$

$=\{$ let prop. $\}$

$$
\begin{aligned}
State\ (\lambda s \to let\ &(v_0 :!: s'_0) = runState\ t\ s\\
&(v :!: s') = runState\ (f\ v_0)\ s'_0)\\
&in\ runState\ (g\ v)\ s')
\end{aligned}
$$

(1)

Luego, del otro lado de la igualdad:

$$\text{t} \ggg= (\lambda x \to \text{(f x)} \ggg= g)$$

$=\{$ $\ggg=$,1 $\}$

$$
\begin{gathered}
State(\lambda s \to let\ (v :!: s') = runState\ t\ s\\
in\ runState((\lambda x \to f\ x \ggg= g)\ v)\ s')
\end{gathered}
$$

$=\{$ $\ggg=$,1 $\}$

$$
\begin{gathered}
State(\lambda s \to let\ (v :!: s') = runState\ t\ s\\
in\ runState((\lambda x \to\\
(State(\lambda s'_0 \to let\ (v :!: s'_0) = runState(f\ x)\ s'_0\\
in\ runState(g\ v_0)\ s'_0))\ v)\ s')
\end{gathered}
$$

$=\{$ β-redex $\}$

$$
\begin{gathered}
State(\lambda s \to let(v :!: s') = runState\ t\ s\\
in\ runState(State(\lambda s'_0 \to let\ (v :!: s'_0) = runState(f\ v)\ s'_0\\
in\ runState(g\ v_0)\ s'_0)\ s')
\end{gathered}
$$

$=\{$ runState . State $\equiv$ id $\}$

$$
\begin{gathered}
State(\lambda s \to let\ (v :!: s') = runState\ t\ s\\
in\ (\lambda s_0 \to let\ (v :!: s'_0) = runState(f\ v)\ s'_0\\
in\ runState(g\ v_0)\ s'_0)\ s')
\end{gathered}
$$

$=\{$ β-redex $\}$

$$
\begin{gathered}
State(\lambda s \to let\ (v :!: s') = runState\ t\ s\\
in\ let\ (v :!: s'_0) = runState(f\ v)\ s'\\
in\ runState(g\ v_0)\ s'_0)
\end{gathered}
$$

$=\{$ let prop. anidado $\}$

$$
\begin{aligned}
State\ (\lambda s \to let\ &(v :!: s') = runState\ t\ s\\
&(v_0 :!: s'_0) = runState\ (f\ v)\ s'\\
&in\ runState\ (g\ v_0)\ s'_0)
\end{aligned}
$$

(2)

Por lo tanto, como (1) = (2). Se tiene que *monad*.3 vale.

- **Propiedad let anidado:**

```
let x = let y = f
in g y
in h x
```

Equivale a: ($\iff$ y $\notin$ FV (h x))

```
let y = f
x = g y
in h x
```

Equivale a:

```
let y = f
in let x = g y
in h x
```

2. Ejercicios 1b a 3

Las mónadas utilizadas en los tres evaluadores están definidas en el archivo Monads.hs. Para cada ejercicio N, se encuentran implementados las instancias monádicas requeridas y el evaluador en el archivo EvalN.hs, donde $N \in \{1, 2, 3\}$.