TP3: Lenguaje Imperativo Simple Monadico

Grillo (G-5811/4), Libonati (L-3256/5), Maiza (M-7116/1)

EJERCICIO 1 a)

Vamos a demostrar que State es una mónada probando las tres leyes de mónadas para la instancia dada.

Monad.1: $return\ a >>= k = k\ a$

$$return\ x >>= f$$

$$= <\text{return.1}>$$

$$State\ (\lambda s \rightarrow (x\ \ :!:\ \ s)) >>= f$$

$$= <(>>=).1>$$

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

$$= <\text{Lema 1: } (runState.State) = (State.runState) = id, \text{ id.1, Def. (.)}>$$

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

$$= <\text{Aplicación}>$$

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

$$= <\text{Def. let}>$$

$$State\ (\lambda s'' \rightarrow runState\ (f\ \ x)\ s'')$$

$$= <\eta\text{-reducción: } (\lambda x \rightarrow\ f\ \ x) = f>$$

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

$$= <\text{Lema 1, id.1, Def. (.)}>$$

$$f\ \ x$$

Monad.2: $m >>= return = m$

$$(State\ h) >>= return$$

$$= <\ (>>=).1\ >$$

$$\begin{aligned}&State(\lambda s \rightarrow let(v\ \ :!:\ \ s') = runState\ (State\ h)\ s\\ &\quad in\ runState\ (return\ v)\ s')\end{aligned}$$

$$= <\text{ Lema 1: runState.State = State.runState = id } >$$

$$\begin{aligned}&State(\lambda s \rightarrow let\ (v\ \ :!:\ \ s') = h\ s\\ &\quad in\ runState\ (return\ v)\ s')\end{aligned}$$

$$= <\text{ Def. return } >$$

$$\begin{aligned}&State(\lambda s \rightarrow let\ (v\ \ :!:\ \ s') = h\ s\\ &\quad in\ runState\ (State(\lambda s'' \rightarrow\ (v\ \ :!:\ \ s'')))\ s')\end{aligned}$$

$$= <\text{ Lema 1 } >$$

$$\begin{aligned}&State(\lambda s \rightarrow let\ (v\ \ :!:\ \ s') = h\ s\\ &\quad in\ (\lambda s'' \rightarrow\ (v\ \ :!:\ \ s''))s')\end{aligned}$$

$$= <\text{ App } >$$

$$\begin{aligned}&State(\lambda s \rightarrow let\ (v\ \ :!:\ \ s') = h\ s\\ &\quad in\ (v\ \ :!:\ \ s'))\end{aligned}$$

$$= <\text{ Def. let } >$$

$$State(\lambda s \rightarrow\ h\ s)$$

$$= <\ \eta\text{-reducción } >$$

$$\text{(State h)}$$

Monad.3: $m >>= (\lambda x \ \rightarrow \ k \ x >>= h) = (m >>= k) >>= h$

$$State \ f >>= (\lambda x \ \rightarrow \ k \ x >>= h)$$

$$= < \ (>>=).1 \ >$$

$$\begin{aligned} State(\lambda s \rightarrow let(v \ \ :! : \ s') &= runState \ (State \ f) \ s \\ in \ runState \ ((\lambda x \ \rightarrow \ k \ x >>= h) \ v) \ s') \end{aligned}$$

$$=< \text{Lema 1: runState.State = State.runState = id} >$$

$$\begin{aligned} State(\lambda s \rightarrow let(v \ \ :! : \ s') &= f \ s \\ in \ runState \ ((\lambda x \ \rightarrow \ k \ x >>= h) \ v) \ s') \end{aligned}$$

$$= < \text{App} >$$

$$\begin{aligned} State(\lambda s \rightarrow let(v \ \ :! : \ s') &= f \ s \\ in \ runState \ (k \ v >>= h) \ s') \end{aligned}$$

$$=< \ (>>=).1 \ >$$

$$\begin{aligned} &State(\lambda s \rightarrow \ let(v \ \ :! : \ s') = f \ s \\ &in \ runState \ (State(\lambda s'' \rightarrow \ let(v' \ \ :! : \ s''') = runState \ (k \ v) \ s'' \\ &\qquad in \ runState \ (h \ v') \ s''')) \ s') \end{aligned}$$

$$= < \text{Lema 1} >$$

$$\begin{aligned} &State(\lambda s \rightarrow \ let \ (v \ \ :! : \ s') = f \ s \\ &in \ (\lambda s'' \rightarrow \ let \ (v' \ \ :! : \ s''') = runState \ (k \ v) \ s'' \\ &\qquad in \ runState \ (h \ v') \ s''') \ s') \end{aligned}$$

Para completar la prueba vamos a comenzar desde el final

$$(State \ f >>= k) >>= h$$

$$=< \ (>>=).1 \ >$$

$$\begin{aligned} State(\lambda s \rightarrow let \ (v \ \ :! : \ s') &= runState \ (State \ f) \ s \\ in \ runState \ (k \ v) \ s') >>= h \end{aligned}$$

$$= < \text{Lema 1} >$$

$$\begin{aligned} State(\lambda s \rightarrow let \ (v \ \ :! : \ s') &= f \ s \\ in \ runState \ (k \ v) \ s') >>= h \end{aligned}$$

$$=< \ (>>=).1 \ >$$

$$\begin{aligned}
State(\lambda s'' \rightarrow \ let \ (v' \ \ :! : \ s''') = runState \ (State(\lambda s \rightarrow \ let(v \ \ :! : \ s') = f \ s \\
in \ runState \ (k \ v) \ s')) \ s'') \\
in \ runState \ (h \ v') \ s''')
\end{aligned}$$

$$= < \text{Lema 1} >$$

$$\begin{aligned}
State(\lambda s'' \rightarrow \ let \ (v' \ \ :! : \ s''') = (\lambda s \rightarrow \ let(v \ \ :! : \ s') = f \ s \\
in \ runState \ (k \ v) \ s') \ s'') \\
in \ runState \ (h \ v') \ s''')
\end{aligned}$$

$$= < \text{App} >$$

$$\begin{aligned}
State(\lambda s'' \rightarrow \ let \ (v' \ \ :! : \ s''') = let \ (v \ \ :! : \ s') = f \ s'' \\
in \ runState \ (k \ v) \ s')) \\
in \ runState \ (h \ v') \ s''')
\end{aligned}$$

$$\begin{aligned}
= < \text{Lema 2: } \lambda s'' \rightarrow \text{let (c :!: d) = let (a :!: b) =} f \ s'' \\
in \ runState \ (k \ a) \ b \\
in \ runState \ (h \ c) \ d \\
= \\
\lambda s'' \rightarrow \ let \ (a :! : b) = f \ s'' \\
(c :! : d) = runState \ (k \ a) \ b \\
in \ runState \ (h \ c) \ d >
\end{aligned}$$

$$\begin{aligned}
State(\lambda s'' \rightarrow \ let \ let \ (v \ \ :! : \ s') = f \ s'' \\
(v' \ \ :! : \ s''') = in \ runState \ (k \ v) \ s' \\
in \ runState \ (h \ v') \ s''')
\end{aligned}$$