Práctica 9.

1. Evalue de modo eager y normal las expresiones **True** $\vee$ 0, **True** $\vee$ $\Delta\Delta$, donde $\Delta = (\lambda x.\, xx)$.

$\text{True} \vee 0$

$\quad \text{True} \Rightarrow_E \text{True}$

$\quad 0 \Rightarrow_E 0$

$\Rightarrow_E \text{True} \vee 0$

$\quad$ (error tipo)

$\text{True} \vee 0$

$\quad \text{True} \Rightarrow_N \text{True}$

$\Rightarrow_N \text{True}$

$\text{True} \vee \Delta\Delta$

$\quad \text{True} \Rightarrow_E \text{True}$

$\quad \Delta\Delta \Rightarrow_E \bot$

$\Rightarrow_E \bot$

$\text{True} \vee \Delta\Delta$

$\quad \text{True} \Rightarrow_N \text{True}$

$\Rightarrow_N \text{true}$

2. Considere la expresión **let** f $\equiv$ λx.**True** **in** f (**True**+0). Recuerde que **let** f $\equiv e$ **in** $e' \doteq (\lambda \text{f}.e')e$
   a) Explique, sin hacer ninguna evaluación, si ese término tiene o no forma canónica en evaluación eager y en evaluación normal.
   b) Construya el árbol de la evaluación para cada uno de esos órdenes.

Reescribamos

$$\text{let } f \equiv \lambda x.\text{True in } f(\text{True}+0) = (\lambda f.\ f(\text{True}+0))\cdot(\lambda x.\text{True})$$

a) En evaluación eager no, ya que True + 0 da error de tipo, en la normal si.

b)

$$\cfrac{(\lambda f.\ f(\text{True}+0)) \Rightarrow_E \lambda f.\ f(\text{True}+0) \qquad \lambda x.\text{True} \Rightarrow_E \lambda x.\text{True} \qquad \cfrac{(\lambda x.\text{True}) \Rightarrow_E (\lambda x.\text{True}) \qquad (\text{True}+0) \Rightarrow_E \text{tyerr} \qquad (\lambda x.\text{True})(x/\text{tyerr}) \Rightarrow_E \text{tyerr}}{(\lambda x.\text{True})\,(\text{True}+0) \Rightarrow_E \text{tyerr}}}{(\lambda f.\ f(\text{true}+0))\,(\lambda x.\text{True}) \Rightarrow_E \text{tyerr}}$$

$$\dfrac{(\lambda f.\ f(True \div 0)) \Rightarrow_N \lambda f.\ f(True \div 0) \qquad \dfrac{(\lambda x.True) \Rightarrow_N (\lambda x.True) \qquad (\lambda x.true)(x/(True \div 0)) \Rightarrow_N True}{(\lambda f.\ f(True \div 0))(f/\lambda x.True) \Rightarrow_N True}}{(\lambda f.\ f(true \div 0))(\lambda x.True) \Rightarrow_N True}$$

3. Extender la semántica de la evaluación para describir el tratamiento de errores. Para esto incorpore las expresiones **error** y **typeerror** como resultados posibles de una evaluación, al mismo nivel que las formas canónicas. Por ejemplo se deberán agregar (entre otras) la reglas:

$$\frac{e \Rightarrow \lfloor i \rfloor \qquad e' \Rightarrow \lfloor 0 \rfloor}{e \div e' \Rightarrow \textbf{error}} \qquad \frac{e \Rightarrow \lfloor b \rfloor \qquad e' \Rightarrow z'}{e \vee e' \Rightarrow \textbf{typeerror}} \, (z' \notin \langle boolcnf \rangle)$$

$$\frac{e \Rightarrow z}{-e \Rightarrow typeerror} \qquad z \notin \langle intcfm \rangle$$

$$\frac{e \Rightarrow z}{\neg e \Rightarrow typeerror} \qquad z \notin \langle boolcfm \rangle$$

$$\frac{e \Rightarrow \lfloor b \rfloor \qquad e' \Rightarrow z'}{e \oplus e' \Rightarrow typeerror} \qquad z' \notin \langle boolcfm \rangle \quad \oplus \in \{\vee, \wedge, \dots\}$$

idem si $e \Rightarrow z$ $\quad z \notin \langle boolcfm \rangle$

$$\frac{e \Rightarrow \lfloor i \rfloor \qquad e' \Rightarrow z'}{e \oplus e' \Rightarrow typeerror} \qquad z' \notin \langle intcfm \rangle \quad \oplus \in \{+, -, /, rem, *, \dots\}$$

idem si $e \Rightarrow z$ $\quad z \notin \langle intcfm \rangle$

$$\frac{e \Rightarrow \lfloor i \rfloor \qquad e' \Rightarrow \lfloor 0 \rfloor}{e \oplus e' \Rightarrow error} \qquad \oplus \in \{rem, /\}$$

$$\frac{e \Rightarrow \varepsilon}{e \oplus e' \Rightarrow \varepsilon} \qquad \varepsilon \in \{error, typeerror\} \quad \oplus \in \{+, -, *, /, \dots\}$$

$$\frac{e \Rightarrow \lfloor i \rfloor \qquad e' \Rightarrow \varepsilon}{e \oplus e' \Rightarrow \varepsilon} \qquad \varepsilon \in \{error, typeerror\} \quad \oplus \in \{+, -, *, /, \dots\}$$

$$\frac{e \Rightarrow \varepsilon}{e \oplus e' \Rightarrow \varepsilon} \qquad \varepsilon \in \{error, typeerror\} \quad \oplus \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow, \dots\}$$

$$\frac{e \Rightarrow \lfloor b \rfloor \qquad e' \Rightarrow \varepsilon}{e \oplus e' \Rightarrow \varepsilon} \qquad \varepsilon \in \{error, typeerror\} \quad \oplus \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow, \dots\}$$

$$\frac{e \Rightarrow \varepsilon}{\neg e \Rightarrow \varepsilon} \qquad \varepsilon \in \{error, typeerror\}$$

$$\frac{e \Rightarrow \varepsilon}{- e \Rightarrow \varepsilon} \qquad \varepsilon \in \{error, typeerror\}$$

$$\frac{e \Rightarrow \varepsilon}{e e' \Rightarrow \varepsilon} \qquad \varepsilon \in \{error, typeerror\}$$

$$\frac{e \Rightarrow_{\varepsilon} \lambda v.e'' \qquad e' \Rightarrow_{\varepsilon} \varepsilon}{e e' \Rightarrow_{\varepsilon} \varepsilon} \qquad \varepsilon \in \{error, typeerror\}$$

$$\frac{e \Rightarrow z}{\text{if } e \text{ then } e' \text{ else } e'' \Rightarrow typeerror} \qquad z \notin \langle boolcfm \rangle$$

$$\frac{e \Rightarrow \varepsilon}{\text{if } e \text{ then } e' \text{ else } e'' \Rightarrow \varepsilon} \qquad \varepsilon \in \{typeerror, error\}$$

$$\frac{e \Rightarrow True \qquad e' \Rightarrow \varepsilon}{\text{if } e \text{ then } e' \text{ else } e'' \Rightarrow \varepsilon} \qquad \varepsilon \in \{typeerror, error\}$$

$$\frac{e \Rightarrow False \qquad e'' \Rightarrow \varepsilon}{\text{if } e \text{ then } e' \text{ else } e'' \Rightarrow \varepsilon} \qquad \varepsilon \in \{typeerror, error\}$$

4. Analice qué regimen de evaluación de subexpresiones ha adoptado en la semántica del ejercicio anterior. Esto es, las subexpresiones (por ejemplo en $e + e'$) se evalúan antes de chequear que los tipos sean correctos, o se obtiene **typeerror** frente a una inconsistencia de tipos, aunque no se hayan terminado de evaluar todas las subexpresiones?
   De reglas que representen la opción no considerada en el ejercicio anterior.

En el caso de las reglas del ejercicio 3 se evalúa siempre la primera expresión y luego la segunda, por ende cualquier error encontrado en la evaluación de la primera expresión se va a propagar.

Para la otra opción solo tenemos el caso en que $e \Rightarrow \varepsilon$ con $\varepsilon \in \{\text{error}, \text{typeerror}\}$ y $e' \Rightarrow \varepsilon'$ con $\varepsilon' \neq \varepsilon$ y $\varepsilon' \in \{\text{error}, \text{typeerror}\}$.

En ese caso debemos escoger cual error toma precedencia, para que sea distinto al anterior demos prioridad a error.

$$\frac{e \Rightarrow \varepsilon \qquad e' \Rightarrow \text{error}}{e \oplus e' \Rightarrow \text{error}} \qquad \begin{array}{l} \varepsilon \in \{\text{typeerror}, \text{error}\} \\ \oplus \in \{\wedge, \vee, \dots, +, -, \times, \backslash, \dots\} \end{array}$$

$$\frac{e \Rightarrow \varepsilon \qquad e' \Rightarrow z}{e \oplus e' \Rightarrow \varepsilon} \qquad \begin{array}{l} \varepsilon \in \{\text{type error}, \text{error}\} \\ z \in \langle boolcfm\rangle \text{ y } \oplus \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow, \dots\} \\ \text{o } z \in \langle intcfm\rangle \text{ y } \oplus \in \{+, -, \times, \backslash, \dots\} \end{array}$$

5. Evalue de modo eager y normal las expresiones $\langle \mathbf{True} + 0, \Delta\Delta\rangle$ y $\langle \Delta\Delta, \mathbf{True} + 0\rangle$.

Normal

$\langle \text{True} + 0, \Delta\Delta\rangle \Rightarrow \langle \text{True} + 0, \Delta\Delta\rangle$

$\langle \Delta\Delta, \text{True} + 0\rangle \Rightarrow \langle \Delta\Delta, \text{True} + 0\rangle$

Eager

$\langle \text{True} + 0, \Delta\Delta\rangle$

$\text{True} \Rightarrow \text{True}$

$0 \Rightarrow 0$

$\text{True} + 0 \Rightarrow \text{type error}$

No se puede evaluar

$\langle \Delta\Delta, \text{True} + 0 \rangle$

$\Delta\Delta \Rightarrow$ No se puede evaluar

6. Para el lenguaje aplicativo normal, reescribir utilizando patrones y **rec**, el término

$$\begin{array}{lrcl} \textbf{letrec} & par & \equiv & \lambda x.\textbf{if } x = 0 \textbf{ then true else } impar(x-1) \\ & impar & \equiv & \lambda x.\textbf{if } x = 0 \textbf{ then false else } par(x-1) \\ \textbf{in } e & & & \end{array}$$

letrec $f \equiv \lambda x . e_0$ in e $=_{def}$ let $f \equiv \text{rec}(\lambda f. \lambda x\, e_0)$ in e

let
par $\equiv$ rec($\lambda f. \lambda x$ if $x=0$ then true else impar $(x-1)$)
impar $\equiv$ rec ($\lambda f. \lambda x$ if $x=0$ then false else par $(x-1)$)
in e

$(\lambda par . \lambda impar . e)$ (rec $(\lambda f. \lambda x \ldots)$) (rec $(\lambda f. \lambda x \ldots)$)

7. De una expresión e tal que esta tenga forma canónica bajo orden normal y que también la tengan las siguientes (infinitas) expresiones: $e.1$, $(e.2).1$, $((e.2).2).1$, etc.

$e =$ rec $(\lambda w \langle 0, w \rangle)$

La forma canónica es $(\lambda w \langle 0, w \rangle)$(rec $(\lambda w \langle 0, w \rangle)) = \langle 0,$ rec $(\lambda w \langle 0, w \rangle)\rangle$

$e.1 \Rightarrow 0$

$(e.2).1 \Rightarrow$ (rec $(\lambda w \langle 0, w \rangle)).1 \Rightarrow 0$

⋮

8. Suponga que e es una expresión cerrada. Considere las siguientes expresiones:

$$\begin{array}{ll} \textbf{letrec } \mathsf{f} \equiv \lambda x.\ \textbf{if } e \textbf{ then } 1 \textbf{ else } \mathsf{f}\ x & \textbf{in } \mathsf{f}\ 0 \\ \textbf{letrec } \mathsf{f} \equiv \lambda x.\ \textbf{if } e \textbf{ then True else } \mathsf{f}\ x & \textbf{in } \mathsf{f}\ 0 + 1 \end{array}$$

Evaluar del modo eager y normal estos programas, considerando por separado los casos $e \Rightarrow$ **true** y $e \Rightarrow$ **false**.

Normal

$e \Rightarrow true$

$\text{letrec } f \equiv \lambda x . \text{ if } e \text{ then } 1 \text{ else } f x \text{ in } f 0 =_{def}$

$\text{let } f \equiv rec(\lambda f. \lambda x. \text{ if } e \text{ then } 1 \text{ else } f x) \text{ in } f 0$

$\text{let } f \equiv rec(\lambda f. \lambda x \text{ if } e \text{ then } 1 \text{ else } f x) \text{ in } f 0 \Rightarrow$

$(\lambda f. f 0)\ (rec(\lambda f. \lambda x \text{ if } e \text{ then } 1 \text{ else } f x))$

$(\lambda f. f 0) \Rightarrow (\lambda f. f 0)$

$(rec(\lambda f. \lambda x \text{ if } e \text{ then } 1 \text{ else } f x)\ 0$ $\quad \to e = (\lambda f. \lambda x \text{ if } e \ldots)$

$rec\ e \Rightarrow (\lambda x \text{ if } e \text{ then } 1 \text{ else } (rec\, e)\, x)$

$\text{if } e \text{ then } 1 \text{ else } (rec\, e)\ 0$

$e \Rightarrow true$

$\Rightarrow 1$

$\Rightarrow 1$

$\Rightarrow 1$

$\text{let } f \equiv rec(\lambda f. \lambda x \text{ if } e \text{ then } 1 \text{ else } f x) \text{ in } f 0 \Rightarrow$

$(\lambda f. f 0)\ (rec(\lambda f. \lambda x \text{ if } e \text{ then } 1 \text{ else } f x))$

$(\lambda f. f 0) \Rightarrow (\lambda f. f 0)$

$(rec(\lambda f. \lambda x \text{ if } e \text{ then } 1 \text{ else } f x)\ 0$

$rec\ e \Rightarrow (\lambda x \text{ if } e \text{ then } 1 \text{ else } (rec\, e)\, x)$

$\text{if } e \text{ then } 1 \text{ else } (rec\, e)\ 0$

$e \Rightarrow false$

$(rec\, e)\ 0$

$(rec\, e) \Rightarrow (\lambda x \text{ if } e \text{ then } 1 \text{ else } (rec\, e)\, x)$

$\text{if } e \text{ then } 1 \text{ else } (rec\, e)\ 0$

$e \Rightarrow false$

$(rec\, e)\ 0$

$\vdots$

No termina la evaluación

Eager

letrec $f \equiv \lambda x.$ if e then 1 else $f\,x$

$(f\ 0)\ /\ (f \to \lambda x. e_0^*)$

$e_0 =$ if e then 1 else $f\,x$

$e_0^* =$ letrec $f \equiv \lambda x. e_0$ in e_0

letrec $f \equiv \lambda x.$ if e then 1 else $f\,x$

$(\lambda x$ letrec $f \equiv \lambda x.($if e then 1 else $f\,x)$ in if e then 1 else $f x)\ 0$

λx letrec $f \equiv \lambda x \ldots \Rightarrow \lambda x$ letrec $f \equiv \lambda x \ldots$

$0 \Rightarrow 0$

letrec $f \equiv \lambda x$ (if e then 1 else $f\,x$) in if e then 1 else $f0$

letrec $f \equiv \lambda x \ldots \Rightarrow ($if e then 1 else $f0)\ /\ (f \mapsto \lambda x. e_0^*)$

$\Rightarrow$ if e then 1 else $(\lambda x. e_0^*)\ 0$

e $\Rightarrow$ true

$1 \Rightarrow 1$

$\Rightarrow 1$

$\Rightarrow 1$

$\Rightarrow 1$

letrec $f \equiv \lambda x.$ if e then 1 else $f\,x$

$(\lambda x$ letrec $f \equiv \lambda x.($if e then 1 else $f\,x)$ in if e then 1 else $f x)\ 0$

λx letrec $f \equiv \lambda x \ldots \Rightarrow \lambda x$ letrec $f \equiv \lambda x \ldots$

$0 \Rightarrow 0$

letrec $f \equiv \lambda x$ (if e then 1 else $f\,x$) in if e then 1 else $f0$

letrec $f \equiv \lambda x \ldots \Rightarrow ($if e then 1 else $f0)\ /\ (f \mapsto \lambda x. e_0^*)$

$\Rightarrow$ if e then 1 else $(\lambda x. e_0^*)\ 0$

e $\Rightarrow$ true

$(\lambda x. e_0^*)\ 0$

$(\lambda x. e_0^*) \Rightarrow (\lambda x. e_0^*)$

$0 \Rightarrow 0$

letrec $f \equiv \lambda x$ (if e then 1 else $f x$) in if e then 1 else $f 0$

$\vdots$

No termina la evaluación.

Normal

let $f \equiv rec(\lambda f. \lambda x$ if e then True else $f x)$ in $f\ 0+1 \Rightarrow$

$(\lambda f. f\, 0+1)$ (rec ($\lambda f. \lambda x$ if e then true else $f x$))

$(\lambda f. f(0+1)) \Rightarrow (\lambda f. f(0+1))$

(rec ($\lambda f. \lambda x$ if e then true else $f x$) (0+1))

rec e $\Rightarrow$ (λx if e then true else (rec e) x)

if e then true else (rec e) (0+1)

e $\Rightarrow$ true

true $\Rightarrow$ true

$\Rightarrow$ true

$\Rightarrow$ true

$\Rightarrow$ true

let $f \equiv rec(\lambda f. \lambda x$ if e then True else $f x)$ in $f\ 0+1 \Rightarrow$

$(\lambda f. f\, 0+1)$ (rec ($\lambda f. \lambda x$ if e then true else $f x$))

$(\lambda f. f(0+1)) \Rightarrow (\lambda f. f(0+1))$

(rec ($\lambda f. \lambda x$ if e then true else $f x$) (0+1))

rec e $\Rightarrow$ (λx if e then true else (rec e) x)

if e then true else (rec e) (0+1)

e $\Rightarrow$ false

(rec e) (0+1)

rec e $\Rightarrow$ (λx if e then true else (rec e) x)

$$\text{if } e \text{ then true else } (rec\ e)\ (0+1)$$

$$e \Rightarrow false$$

$$(rec\ e)\ (0+1)$$

$$\vdots$$

No termina la evaluación

Eager

$$\text{letrec } f \equiv \lambda x.\ \text{if } e \text{ then true else } f\,x$$

$$(\lambda x\ \text{letrec } f \equiv \lambda x.(\text{if } e \text{ then true else } f\,x) \text{ in if } e \text{ then true else } f\,x)\ (0+1)$$

$$\lambda x\ \text{letrec } f \equiv \lambda x \ldots \Rightarrow \lambda x\ \text{letrec } f \equiv \lambda x \ldots$$

$$(0+1)$$

$$0 \Rightarrow 0$$

$$1 \Rightarrow 1$$

$$\Rightarrow 1$$

$$\text{letrec } f \equiv \lambda x\ (\text{if } e \text{ then true else } f\,x) \text{ in if } e \text{ then true else } f\ 1$$

$$\text{letrec } f \equiv \lambda x \ldots \Rightarrow (\text{if } e \text{ then true else } f\,1)\ /\ (f \mapsto \lambda x.e_0^*)$$

$$\Rightarrow \text{if } e \text{ then true else } (\lambda x.e_0^*)\ 1$$

$$e \Rightarrow true$$

$$true \Rightarrow true$$

$$\Rightarrow true$$

$$\Rightarrow true$$

$$\Rightarrow true$$

$$\Rightarrow true$$

letrec $f \equiv \lambda x.$ if e then true else $f\,x$

$(\lambda x$ letrec $f \equiv \lambda x.($if e then true else $f\,x)$ in if e then true else $f x)\ (0+1)$

λx letrec $f = \lambda x \ldots \Rightarrow \lambda x$ letrec $f \equiv \lambda x \ldots$

$(0+1)$

$0 \Rightarrow 0$

$1 \Rightarrow 1$

$\Rightarrow 1$

letrec $f \equiv \lambda x$ (if e then true else $f\,x$) in if e then true else $f\ 1$

letrec $f \equiv \lambda x \ldots \Rightarrow$ (if e then true else $f 1$) / ($f \mapsto \lambda x.e_0^*$)

$\Rightarrow$ if e then true else $(\lambda x.e_0^*)\ 1$

$e \Rightarrow$ false

$(\lambda x.e_0^*)\ 1$

$(\lambda x.e_0^*) \Rightarrow (\lambda x.e_0^*)$

$1 \Rightarrow 1$

letrec $f \equiv \lambda x$ (if e then true else $f x$) in if e then true else $f\ 1$

$\vdots$

No termina la evaluación

9. Decida si la siguiente afirmación *Mmm*? es cierta o no y justifique su respuesta: "Si $e \Rightarrow_E z$, entonces toda subexpresión e' de e tiene forma canónica".

Falso $\quad \lambda x.\Delta\Delta \Rightarrow_E \lambda x.\Delta\Delta$ pero $\Delta\Delta$ no tiene forma canónica