Ej 2

a) *Reglas de tipado*

Agrego una regla de tipado para cada termino agregado a la extension

$$\frac{\Gamma \;\rhd\; M : \sigma}{\Gamma \;\rhd\; \text{MAT}(M) : \text{MatInf}_\sigma}\text{(T-MAT)}$$

$$\frac{\Gamma \;\rhd\; M : \text{MatInf}_\sigma \qquad \Gamma \;\rhd\; P : \sigma \qquad \Gamma \;\rhd\; N, O : Nat}{\Gamma \;\rhd\; M[N][O] \leftarrow P : \text{MatInf}_\sigma}\text{(T-MatSet)}$$

$$\frac{\Gamma \;\rhd\; M : \text{MatInf}_\sigma \qquad \Gamma \;\rhd\; N, O : Nat}{\Gamma \;\rhd\; M[N][O] : \sigma}\text{(T-MatGet)}$$

$$\frac{\Gamma \;\rhd\; M : \text{MatInf}_\sigma \qquad \Gamma \;\rhd\; N : \sigma \to \tau}{\Gamma \;\rhd\; map(M, N) : \text{MatInf}_\tau}\text{(T-MatMap)}$$

b) *Demostrar el siguiente juicio de tipado o explicar por que no es derivable*

$$\{i : Nat, m : \text{MatInf}_{Nat}\} \;\rhd\; map((m[i][i] \leftarrow 0), \lambda x : Nat.isZero(x)) : \text{MatInf}_{Bool}$$

Intuitivamente deberia ser derivable, ya que mapea una matriz infinita de naturales a una matriz de booleanos que indica si cada uno era cero. Intento de demostrarlo

$$\dfrac{\dfrac{\dfrac{\dfrac{\checkmark}{m : \text{MatInf}_{Nat} \in \Gamma}}{\Gamma \;\rhd\; m : \text{MatInf}_{Nat}}\text{(T-Var)} \qquad \dfrac{\dfrac{\checkmark}{i : Nat \in \Gamma}}{\Gamma \;\rhd\; i : Nat}\text{(T-Var)} \qquad \dfrac{\checkmark}{\Gamma \;\rhd\; 0 : Nat}\text{(T-Zero)}}{\Gamma \;\rhd\; m[i][i] \leftarrow 0 : \text{MatInf}_{Nat}}\text{(T-MatSet)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\checkmark}{x : Nat \in \Gamma \cup \{x : Nat\}}}{\Gamma \cup \{x : Nat\} \;\rhd\; x : Nat}\text{(T-Var)}}{\Gamma \cup \{x : Nat\} \;\rhd\; isZero(x) : Bool}\text{(T-IsZero)}}{\Gamma \;\rhd\; \lambda x : Nat.isZero(x) : Nat \to Bool}\text{(T-Abs)}}{\Gamma = \{i : Nat, m : \text{MatInf}_{Nat}\} \;\rhd\; map((m[i][i] \leftarrow 0), \lambda x : Nat.isZero(x)) : \text{MatInf}_{Bool}}\text{(T-MatMap)}$$

c) *Indicar formalmente cómo se modifica el conjunto de valores, o explicar por qué no se modifica.*

Es necesario agregar las matrices como valor, la idea va a ser similar a como se construyen los naturales. $MAT(M)$ va a ser un valor con una matriz con todos Ms, y se va construir con $V[X][Y] \leftarrow M$. Esto quiere decir que, por ejemplo, $((MAT(0)[0][0] \leftarrow 1)[1][1] \leftarrow 2)[2][2] \leftarrow 3$ seria un valor valido. El conjunto de valores entonces sería

$$V, X, Y ::= \ldots \mid MAT(M) \mid V[X][Y] \leftarrow M$$

d) *Dar la semántica operacional de a un paso para la extensión.*

$$\frac{M_1 \to M_1'}{M_1[X][Y] \leftarrow M_2 \to M_1'[X][Y] \leftarrow M_2}\text{(E-MatSetC)}$$

$$\frac{N \to N'}{M_1[N][O] \leftarrow M_2 \to M_1[N'][O] \leftarrow M_2}\text{(E-MatSetXC)}$$

$$\frac{O \to O'}{M_1[N][O] \leftarrow M_2 \to M_1[N][O'] \leftarrow M_2}\text{(E-MatSetYC)}$$

$$\frac{}{(V[I][J] \leftarrow M)[X][Y] \to V[X][Y]}\text{(E-MatGetNoMatch)}$$

$$\frac{N \to N'}{M[N][O] \to M[N'][O]}\text{(E-MatGetXC)}$$

$$\frac{O \to O'}{M[X][O] \to M[X][O']}\text{(E-MatGetYC)}$$

$$\frac{M \to M'}{M[X][Y] \to M'[X][Y]}\text{(E-MatGetC)}$$

$$\frac{M \to M'}{(V[X][Y] \leftarrow M)[X][Y] \to (V[X][Y] \leftarrow M')[X][Y]}\text{(E-MatGetSetC)}$$

$$\frac{M \to M'}{(\text{MAT}(M))[X][Y] \to (\text{MAT}(M'))[X][Y]}\text{(E-MatGetMATC)}$$

$$\frac{}{(\text{MAT}(V))[X][Y] \to V}\text{(E-MatGetMAT)}$$

$$\frac{}{(V[X][Y] \leftarrow W)[X][Y] \to W}\text{(E-MatGetSet)}$$

$$\frac{F \to F'}{map(M, F) \to map(M, F')}\text{(E-MatMapCF)}$$

$$\frac{M \to M'}{map(M, F) \to map(M', F)}\text{(E-MatMapCM)}$$

$$\frac{}{map(MAT[M], F) \to MAT(FM)}\text{(E-MatMapMAT)}$$

$$\frac{}{map(V[X][Y] \leftarrow M, F) \to map(V, F)[X][Y] \leftarrow F\ M}\text{(E-MatMapSet)}$$

TODO: para matcheo de indices no vale poner la misma letra, hay que preguntar por igualdad sintactica con $\equiv$, como dice el enunciado.

e) *Mostrar como se reducen paso a paso*

   I) $(MAT(\lambda y : Bool.y)[succ(0)][0] \leftarrow \lambda x : Bool.\text{true})[0][succ(0)]$

$$\begin{aligned}
&(MAT(\lambda y : Bool.y)[succ(0)][0] \leftarrow \lambda x : Bool.\text{true})[0][succ(0)] \\
&\to_{\text{(E-MatGetNoMatch)}} (MAT(\lambda y : Bool.y))[0][succ(0)] \\
&\to_{\text{(E-MatGetMAT)}} \lambda y : Bool.y
\end{aligned}$$

II) $((\lambda x : Bool.\text{MAT}(x)[0][0] \leftarrow false)\ true)[0][pred(succ(0)]$

$$
\begin{aligned}
&((\lambda x : Bool.\text{MAT}(x)[0][0] \leftarrow false)\ true)[0][pred(succ(0)]\\
&\rightarrow_{\text{(E-MatGetYC), (E-PredSucc)}} ((\lambda x : Bool.\text{MAT}(x)[0][0] \leftarrow false)\ true)[0][\mathbf{0}]\\
&\rightarrow_{\text{(E-MatGetC, E-AppAbs)}} (\text{MAT}(\boldsymbol{true})[0][0] \leftarrow false)[0][\mathbf{0}]\\
&\rightarrow_{\text{(E-MatGetSet)}} false
\end{aligned}
$$

III) $map(\text{MAT}(0)[0][0] \leftarrow succ(0), \lambda x : Nat.isZero(x))[0][0]$

$$
\begin{aligned}
&map(\text{MAT}(0)[0][0] \leftarrow succ(0), \lambda x : Nat.isZero(x))[0][0]\\
&\rightarrow_{\text{(E-MatGetC, E-MatMapSet)}} map(\text{MAT}(0), \lambda x : Nat.isZero(x))[0][0] \leftarrow (\lambda x : Nat.isZero(x))\ succ(0))[0][0]\\
&\rightarrow_{\text{(E-MatGetC, E-MatMapMAT)}} \text{MAT}((\lambda x : Nat.isZero(x))\ 0)[0][0] \leftarrow (\lambda x : Nat.isZero(x))\ succ(0))[0][0]\\
&\rightarrow_{\text{(E-MatGetSetC, E-AppAbs)}} \text{MAT}((\lambda x : Nat.isZero(x))\ 0)[0][0] \leftarrow isZero(succ(0)))[0][0]\\
&\rightarrow_{\text{(E-MatGetSetC, E-IsZeroSucc)}} \text{MAT}((\lambda x : Nat.isZero(x))\ 0)[0][0] \leftarrow \boldsymbol{false})[0][0]\\
&\rightarrow_{\text{(E-MatGetSet, E-IsZeroSucc)}} false
\end{aligned}
$$