$\boxed{I, \Gamma \vdash_p pgm : \tau}$ Program Typing

$$\frac{I; \Gamma \vdash_{tm} expr : \tau}{I, \Gamma \vdash_p expr : \tau} \textsc{Expression}$$

$$\frac{I, \Gamma \vdash_i impl : I' \qquad I', \Gamma \vdash_p pgm : \tau}{I, \Gamma \vdash_p impl; pgm : \tau} \textsc{implicit}$$

$$\frac{I, \Gamma \vdash_d data : \Gamma' \qquad I, \Gamma' \vdash_p pgm : \tau}{I, \Gamma \vdash_p impl; pgm : \tau} \textsc{data}$$

$\boxed{I, \Gamma \vdash_i impl : I'}$ Building the Implicit Environment

$$\frac{I; \Gamma \vdash_{tm} expr : \tau_1 \to \tau_2}{I, \Gamma \vdash_i implicit\ i : \tau_1 \rightsquigarrow \tau_2 = expr : I, i}$$

$\boxed{I, \Gamma \vdash_d impl : I'}$ Building the Typing Environment

$$I, \Gamma \vdash_d (data\ T\ a = K\ \overline{\tau}) : \Gamma, T, (K : \forall a. \overline{\tau} \to T\ a)$$

$\boxed{I; \Gamma \vdash_{tm} e : \tau}$

$$\frac{I; \Gamma \vdash_{tm} e : \tau; *; *; e'}{I; \Gamma \vdash_{tm} e : \tau}$$

$\boxed{I; \Gamma \vdash_{tm} e : \tau; E; Y; e'}$ Type Inference and Partial Translation

$$\frac{(x : \tau) \in \Gamma}{I; \Gamma \vdash_{tm} x : \tau; \bullet; \bullet; x} \textsc{Var} \qquad \frac{(K : \forall a. \tau) \in \Gamma}{I; \Gamma \vdash_{tm} K : [a \mapsto \tau']\tau; \bullet; \bullet; K} \textsc{Constr}$$

$$\frac{I; \Gamma, x : a \vdash_{tm} e : \tau; E; Y; e'}{I; \Gamma \vdash_{tm} \lambda x.e : a \to \tau; E; Y; e'} \textsc{Abstraction}$$

$$\frac{\begin{array}{c} I; \Gamma \vdash_{tm} e_1 : \tau_1; E_1; Y_1; e_1' \\ I; \Gamma \vdash_{tm} e_2 : \tau_2; E_2; Y_2; e_2' \qquad fresh\ j \end{array}}{I; \Gamma \vdash_{tm} e_1 e_2 : a; E_1 + E_2 + (\tau_1 \sim b \to a); Y_1 + Y_2 + I \vDash_i j : \tau_2 \rightsquigarrow b; e_1'(j\ e_2')} \textsc{Application}$$

$$\frac{\begin{array}{c} (K : \forall a. \overline{\tau} \to T\ a) \in \Gamma \qquad I; \Gamma \vdash_{tm} e_1 : \tau_1; E_1; Y_1; e_1' \\ I; \Gamma, \overline{x} : [a \mapsto b]\overline{\tau} \vdash_{tm} e_2 : \tau_2; E_2; Y_2; e_2' \qquad fresh\ j \end{array}}{I; \Gamma \vdash_{tm} case\ e_1\ of\ (K\ x) \to e_2 : \tau_2; E_1 + E_2; Y_1 + Y_2 + I \vDash_i j : \tau_1 \rightsquigarrow T\ b; case\ (j\ e_1')\ of\ (K\ x) \to e_2'} \textsc{Case}$$

$\boxed{I \vDash_i j : \tau_1 \rightsquigarrow \tau_2; \Theta; \varphi}$ Implicit Conversion Resolution and Subst

$$\frac{\bullet; I \vDash_i j : \tau_1 \rightsquigarrow \tau_2; \Theta; \varphi}{I \vDash_i j : \tau_1 \rightsquigarrow \tau_2; \Theta; \varphi} \text{ Aux}$$

$\boxed{\overline{\tau}; I \vDash_i j : \tau_1 \rightsquigarrow \tau_2; \Theta; \varphi}$ Implicit Conversion Resolution with Loop Detection

$$\frac{[fv(\tau_1) \mapsto \overline{\tau_{1i}}]\tau_1 = [fv(\tau_2) \mapsto \overline{\tau_{2i}}]\tau_2}{\overline{\tau}; I \vDash_i j : \tau_1 \rightsquigarrow \tau_2; [j \mapsto id]; [fv(\tau_1) \mapsto \overline{\tau_{1i}}, fv(\tau_2) \mapsto \overline{\tau_{2i}}]} \text{ Unification}$$

$$\frac{\begin{array}{c}(\forall \overline{a}.Cond \Rightarrow \tau_1' \rightsquigarrow \tau_3' = expr_{fst}) \in I \\ \tau_1 = [\overline{a} \mapsto \overline{\tau_a}]\tau_1' \qquad \tau_3 = [\overline{a} \mapsto \overline{\tau_a}]\tau_3' \qquad \tau_3 \notin \overline{\tau} \\ \forall\ cond_i \in [\overline{a} \mapsto \overline{\tau_a}]Cond\ :\ (I \vDash_i cond_i : \tau_{i1} \rightsquigarrow \tau_{i2}; \Theta_i; \varphi_i) \\ \overline{\tau}, \tau_1; I \vDash_i conv_{rest} : (\overline{\varphi_i}\ \tau_3) \rightsquigarrow \tau_2; \Theta_{rest}; \varphi_{rest}\end{array}}{\overline{\tau}; I \vDash_i conv : \tau_1 \rightsquigarrow \tau_2; [conv \mapsto (\Theta_{rest}\ conv_{rest}) \circ (\overline{\Theta_i}\ ([\overline{a} \mapsto \overline{\tau_a}]expr_{fst}))]; \varphi_{rest} \circ \overline{\varphi_i} \circ [\overline{a} \mapsto \overline{\tau_a}]} \text{ Transitivity}$$

* ::= If all the constraints in E and Y have been satisfied.