Type Judgements

$$\Gamma \vdash x : \Gamma(x) \tag{VAR}$$

$$\Gamma \vdash n : \text{num} \tag{NUM}$$

$$\frac{\Gamma \vdash e_1 : \text{num} \qquad \Gamma \vdash e_2 : \text{num}}{\Gamma \vdash (\text{op}\ e_1\ e_2) : \text{num}} \tag{ARITH}$$

$$\Gamma \vdash \mathit{true} : \text{bool} \tag{TRUE}$$

$$\Gamma \vdash \mathit{false} : \text{bool} \tag{FALSE}$$

$$\frac{\Gamma \vdash e_c : \text{bool} \qquad \Gamma \vdash e_t : \tau \qquad \Gamma \vdash e_f : \tau}{\Gamma \vdash (\text{if}\ e_c\ e_t\ e_f) : \tau} \tag{IF}$$

$$\frac{\Gamma, x : \tau_a \vdash e : \tau}{\Gamma \vdash (\lambda x.e) : \tau_a \to \tau} \tag{λ}$$

$$\frac{\Gamma \vdash e_f : \tau_a \to \tau \qquad \Gamma \vdash e_a : \tau_a}{\Gamma \vdash (e_f\ e_a) : \tau} \tag{APP}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (\text{pair}\ e_1\ e_2) : \tau_1 \times \tau_2} \tag{PAIR}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash (\text{fst}\ e) : \tau_1} \tag{FST}$$

$$\frac{\Gamma \vdash e : \tau_1 \times \tau_2}{\Gamma \vdash (\text{snd}\ e) : \tau_2} \tag{SND}$$

$$\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash (\text{inl}\ e) : \tau_1 + \tau_2} \tag{INL}$$

$$\frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash (\text{inr}\ e) : \tau_1 + \tau_2} \tag{INR}$$

$$\frac{\Gamma \vdash e : \tau_1 + \tau_2 \qquad \Gamma \vdash e_1 : \tau_1 \to \tau \qquad \Gamma \vdash e_2 : \tau_2 \to \tau}{\Gamma \vdash (\text{match}\ e\ e_1\ e_2) : \tau} \tag{MATCH}$$

To extend these rules to support let-polymorphism, we define the concept of a *type closure* for a type τ and type environment Γ, written $\overline{\Gamma}(\tau)$, which is a tuple containing the type and the set of all type variables free in the type that are not known to Γ:

$$\overline{\Gamma}(\tau) = \langle \tau, \text{FreeTypeVars}(\tau) \setminus \text{FreeTypeVars}(\Gamma) \rangle \tag{TYPE-CLOSURE}$$

Inference of a *let* form results in the creation of a type closure for the inferred type of the bound expression and the environment used to infer that type:

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma, x : \overline{\Gamma}(\tau_1) \vdash e : \tau}{\Gamma \vdash (\text{let } x = e_1 \ in \ e) : \tau} \tag{PLET}$$

When a lookup returns a type closure, the closure must be instantiated. This involves generating a fresh type variable for each free variable in the closure and substituting the fresh one for the original. (This allows the polymorphic value to be used independently in different contexts.)

$$\frac{\Gamma(x) = \langle \tau, \{\alpha_1, \ldots, \alpha_n\} \rangle}{\Gamma \vdash x : \tau[\alpha_1 \leftarrow \beta_1, \ldots, \alpha_n \leftarrow \beta_n]} \tag{PVAR}$$