An OCaml Type-System for Typedtrees

Pierrick Couderc   Michel Mauny   Fabrice Le Fessant

February 21, 2017

1 Type checking core-OCaml expressions

$$\textsc{Const}\ \frac{c \in domain(\tau)}{\Gamma, \Phi \vdash \langle c : \tau \rangle} \qquad \textsc{Var}\ \frac{\Gamma, \Phi, \Sigma \vdash \tau \leqslant C.\Gamma.Values(x) \Rightarrow \theta}{\Gamma, \Phi \vdash \langle x : \tau \rangle}$$

$$\textsc{Abs}\ \frac{\begin{array}{c}\Gamma, \Phi \vdash \tau \prec \tau_d \to \tau_{cd} \qquad \forall i.\ \Gamma, \Phi, \Sigma, \Sigma \vdash \langle p_i : \tau_i \rangle \Rightarrow (\overline{v_i : \tau_{v_i}}), (\overline{\tau_{\exists_i}}), \Phi_i \\ \forall i.\ \Gamma, \Phi \vdash \tau_d \equiv \tau_i \qquad \forall i.\ let\ \Gamma_i = \Gamma \oplus_{\mathcal{V}} (\overline{v_i : \tau_{v_i}}) \oplus_{\mathcal{T}} (\overline{\tau_{\exists_i}}) \\ \forall i.\ \Gamma_i, \Phi_i \vdash \langle e_i : \tau'_i \rangle \qquad \forall i.\ \Gamma, \Phi \vdash \tau'_i\ wf \qquad \forall i.\ \Gamma_I, \Phi_i \vdash \tau_{cd} \equiv \tau'_i\end{array}}{\Gamma, \Phi \vdash \langle \mathtt{function}\ \overline{\mid p \to e}\ : \tau \rangle}$$

$$\textsc{Abs-Label}\ \frac{\begin{array}{c}\Gamma, \Phi \vdash \tau \prec l' : \tau_d \to \tau_{cd} \\ l = l' \qquad \Gamma, \Phi, \Sigma, \Sigma \vdash \langle p : \tau_{arg} \rangle \Rightarrow (\overline{v : \tau_v}), (\overline{\tau_\exists}), \Phi' \\ \Gamma, \Phi \vdash \tau_d \equiv \tau_{arg} \qquad let\ \Gamma' = \Gamma \oplus_{\mathcal{V}} (\overline{v : \tau_v}) \oplus_{\mathcal{T}} (\overline{\tau_\exists}) \\ \Gamma', \Phi' \vdash \langle e : \tau_{res} \rangle \qquad \Gamma, \Phi \vdash \tau_{res}\ wf \qquad \Gamma', \Phi' \vdash \tau_{cd} \equiv \tau_{res}\end{array}}{\Gamma, \Phi \vdash \langle \mathtt{function} \sim l : p \to e : \tau \rangle}$$

$$\textsc{App}\ \frac{\begin{array}{c}\Gamma, \Phi \vdash \langle e_1 : \tau_1 \rangle \qquad \Gamma, \Phi \vdash \langle e_2 : \tau_2 \rangle \\ \Gamma, \Phi \vdash \tau_1 \prec \tau_d \to \tau_{cd} \qquad \Gamma, \Phi \vdash \tau_2 \equiv \tau_d \qquad \Gamma, \Phi \vdash \tau_{cd} \equiv \tau\end{array}}{\Gamma, \Phi \vdash \langle e_1\ e_2 : \tau \rangle}$$

$$\textsc{Construct}\ \frac{\begin{array}{c}\forall i.\ \Gamma, \Phi \vdash \langle e_i : \tau_i \rangle \\ let\ (\tau_{arg_0}, \ .. , \tau_{arg_n}, \tau_{constr}) = find\_constructor(\Gamma, \Phi, T, \tau) \\ \Gamma, \Phi, \Sigma \vdash \tau_0 \leqslant \tau_{arg_0} \Rightarrow \theta_0 \qquad \forall i_{\geqslant 1}.\ \Gamma, \Phi, \theta_{i-1} \vdash \tau_i \leqslant \tau_{arg_i} \Rightarrow \theta_i \\ let\ \tau_{inst} = \theta_n(\tau_{constr}) \qquad \Gamma, \Phi \vdash \tau_{inst} \equiv \tau\end{array}}{\Gamma, \Phi \vdash \langle T(e_0,\ .. , e_n) : \tau \rangle}$$

$$\textsc{Let}\ \frac{\begin{array}{c}\forall i.\ \Gamma, \Phi, \Sigma, \Sigma \vdash \langle p_i : \sigma_i \rangle \Rightarrow (\overline{v_i : \sigma_{v_i}}), (\overline{\tau_{\exists_i}}) \\ let\ \mathcal{V}_p = (\overline{v_0 : \sigma_{v_0}}) \uplus .. \uplus (\overline{v_n : \sigma_{v_n}}) \qquad let\ \mathcal{T}_\exists = (\overline{\tau_{\exists_0}}) \uplus .. \uplus (\overline{\tau_{\exists_0}}) \\ \forall i.\ \Gamma, \Phi \vdash \langle e_i : \tau_i \rangle \qquad \forall i.\ \Gamma, \Phi, \Sigma \vdash \tau_i \leqslant \sigma_i \Rightarrow \theta_i \\ \forall i.\ check\_gen(\Gamma, \Phi, \sigma_i, e_i) \qquad let\ \Gamma' = \Gamma \oplus_{\mathcal{V}} \mathcal{V}_p \oplus_{\mathcal{T}} \mathcal{T}_\exists \\ \Gamma', \Phi_n \vdash \langle e' : \tau' \rangle \qquad \Gamma, \Phi \vdash \tau'\ wf \qquad \Gamma', \Phi_n \vdash \tau \equiv \tau'\end{array}}{\Gamma, \Phi \vdash \langle \mathtt{let}\ \overline{p = e}\ \ \mathtt{in}\ e' : \tau \rangle}$$

$$\textsc{Letrec}\ \frac{\begin{array}{c}\Gamma, \Phi, \Sigma, \Sigma \vdash \langle p_0 : \sigma_0 \rangle \Rightarrow (\overline{v_0 : \sigma_{v_0}}), \Phi_0 \\ \forall i_{\geqslant 1}.\ \Gamma, \Phi_{i-1}, \Sigma \vdash \langle p_i : \sigma_i \rangle \Rightarrow (\overline{v_i : \sigma_{v_i}}), \Phi_i \\ let\ \mathcal{V}_p = (\overline{v_0 : \sigma_{v_0}}) \uplus .. \uplus (\overline{v_n : \sigma_{v_n}}) \qquad let\ \mathcal{T}_\exists = (\overline{\tau_{\exists_0}}) \uplus .. \uplus (\overline{\tau_{\exists_0}}) \\ let\ \Gamma' = \Gamma \oplus_{\mathcal{V}} \mathcal{V}_p \oplus_{\mathcal{T}} \mathcal{T}_\exists \qquad \forall i.\ \Gamma', \Phi_n \vdash \langle e_i : \tau_i \rangle \\ \forall i.\ \Gamma, \Phi, \Sigma \vdash \tau_i \leqslant \sigma_i \Rightarrow \theta_i \qquad \forall i.\ check\_gen(\Gamma, \Phi, \sigma_i, e_i) \\ \Gamma', \Phi_n \vdash \langle e' : \tau' \rangle \qquad \Gamma, \Phi \vdash \tau'\ wf \qquad \Gamma', \Phi' \vdash \tau \equiv \tau'\end{array}}{\Gamma, \Phi \vdash \langle \mathtt{let\ rec}\ \overline{p = e}\ \mathtt{in}\ e' : \tau \rangle}$$

2 Pattern typechecking rules

$$\textsc{Pat-Const}\ \frac{c \in domain(\tau)}{\Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle c : \tau \rangle \Rightarrow \mathcal{V}, \mathcal{T}, \Phi}$$

$$\textsc{Pat-Wildcard}\ \Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle \_ : \tau \rangle \Rightarrow \mathcal{V}, \mathcal{T}, \Phi$$

$$\textsc{Pat-Var}\ \frac{v \notin \mathcal{V} \qquad let\ \mathcal{V}' = \mathcal{V} \oplus (v, \tau)}{\Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle v : \tau \rangle \Rightarrow \mathcal{V}', \mathcal{T}, \Phi}$$

$$\textsc{Pat-Tuple}\ \frac{\begin{array}{c}\Gamma, \Phi \vdash \tau \prec \tau_{p_0} *\ ..\ * \tau_{p_n} \qquad \Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle p_0 : \tau_0 \rangle \Rightarrow \mathcal{V}_0, \mathcal{T}_0 \\ \forall i_{\geqslant 1}.\ \Gamma, \Phi_{i-1}, \mathcal{V}_{i-1}, \mathcal{T}_{i-1} \vdash p_i : \tau_i \Rightarrow \mathcal{V}_i, \mathcal{T}_i, \Phi_i \qquad \forall i.\ \Gamma, \Phi_i \vdash \tau_{p_i} \equiv \tau_i\end{array}}{\Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle p_0,\ ..\ , p_n : \tau \rangle \Rightarrow \mathcal{V}_n, \mathcal{T}_n, \Phi_n}$$

$$\textsc{Pat-Or}\ \frac{\begin{array}{c}\Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle p_1 : \tau_1 \rangle \Rightarrow \mathcal{V}_1, \mathcal{T}_1 \Phi_1 \\ \Gamma, \Phi_1, \mathcal{V} \vdash \langle p_2 : \tau_2 \rangle \Rightarrow \mathcal{V}_2, \Phi_2 \qquad \Gamma, \Phi_2 \vdash \tau_1 \equiv \tau_2 \\ \Gamma, \Phi_2 \vdash \tau_2 \equiv \tau \qquad \Gamma, \Phi_2 \vdash \mathcal{V}_1 \equiv \mathcal{V}_2 \qquad \Gamma, \Phi_2 \vdash \mathcal{T}_1 \equiv \mathcal{T}_2\end{array}}{\Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle p_1 \mid p_2 : \tau \rangle \Rightarrow \mathcal{V}_2, \mathcal{T}_2, \Phi_2}$$

$$\textsc{Pat-Construct}\ \frac{\begin{array}{c}\Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle p_0 : \tau_0 \rangle \Rightarrow \mathcal{V}_0, \mathcal{T}_0, \Phi_0 \\ \forall i_{\geqslant 1}.\ \Gamma, \Phi_{i-1}, \mathcal{V}_{i-1}, \mathcal{T}_{i-1} \vdash \langle p_i : \tau_i \rangle \Rightarrow \mathcal{V}_i, \mathcal{T}_i, \Phi_i \\ let\ (\tau_{arg_0},\ ..\ , \tau_{arg_n}, \tau_{constr}, generalized) = find\_constructor(\Gamma, \Phi, T, \tau) \\ let\ \mathcal{T}_{args} = existential\_types((\tau_0 *\ ..\ * \tau_n), \tau_{constr}, (\tau_{arg_0} *\ ..\ * \tau_{arg_n}), generalized) \\ \Gamma, \Phi_n, \Sigma \vdash \tau_0 \leqslant \tau_{arg_0} \Rightarrow \theta_0 \qquad \forall i_{\geqslant 1}.\ \Gamma, \Phi_n, \theta_{i-1} \vdash \tau_i \leqslant \tau_{arg_i} \Rightarrow \theta_i \\ \forall i.\ let\ \Phi_{p_i} = add\_equations((\Gamma, \Phi_{p_{i-1}}), \tau_i, \tau_{arg_i}) \\ let\ \Phi_{ret} = add\_equations((\Gamma, \Phi_{p_n}), \tau, \tau_{constr}) \\ let\ \tau_{inst} = \theta_n(\tau_{constr}) \qquad \Gamma, \Phi_{p_n} \vdash \tau_{inst} \equiv \tau\end{array}}{\Gamma, \Phi, \mathcal{V}, \mathcal{T} \vdash \langle T(p_0,\ ..\ , p_n) : \tau \rangle \Rightarrow \mathcal{V}_n, \mathcal{T}_n, \Phi_{ret}}$$

3 Type instanciation rules

$$\textsc{Inst-Var-Unbound}\ \frac{'a \notin \theta}{\Gamma, \Phi, \theta \vdash\ \tau \leqslant\ 'a \Rightarrow \theta \oplus ['a \rightarrow \tau]}$$

$$\textsc{Inst-Var-Bound}\ \frac{'a \in \theta \qquad \Gamma, \Phi, \theta \vdash \tau_{'a} \equiv \tau}{\Gamma, \Phi, \theta \oplus ['a \rightarrow \tau_{'a}] \vdash\ \tau \leqslant\ 'a \Rightarrow \theta \oplus ['a \rightarrow \tau_{'a}]}$$

$$\textsc{Inst-Var-Generalized}\ \frac{generalized('a_1) \implies generalized('a_2)}{\Gamma, \Phi, \theta \vdash\ 'a_1 \leqslant\ 'a_2 \Rightarrow \theta \oplus ['a_2 \rightarrow' a_1]}$$

$$\textsc{Inst-Fun}\ \frac{l_1 = l_1' \qquad \Gamma, \Phi, \theta \vdash \tau_1 \leqslant \tau_1' \Rightarrow \theta_1 \qquad \Gamma, \Phi, \theta_1 \vdash \tau_2 \leqslant \tau_2' \Rightarrow \theta_2}{\Gamma, \Phi, \theta \vdash (l_1 : \tau_1) \rightarrow \tau_2 \leqslant (l_1' : \tau_1') \rightarrow \tau_2' \Rightarrow \theta_2}$$

$$\textsc{Inst-Tuple}\ \frac{\Gamma, \Phi, \theta \vdash \tau_0 \leqslant \tau_0' \Rightarrow \theta_0 \qquad \forall i_{\geqslant 1}.\ \Gamma, \Phi, \theta_{i-1} \vdash \tau_i \leqslant \tau_i' \Rightarrow \theta_i}{\Gamma, \Phi, \theta \vdash \tau_0 * .. * \tau_n \leqslant \tau_0' * .. * \tau_n' \Rightarrow \theta_n}$$

$$\textsc{Inst-Construct}\ \frac{\Gamma, \Phi, \theta \vdash \tau_0 \leqslant \tau_0' \Rightarrow \theta_0 \qquad \forall i_{\geqslant 1}.\ \Gamma, \Phi, \theta_{i-1} \vdash \tau_i \leqslant \tau_i' \Rightarrow \theta_i}{\Gamma, \Phi, \theta \vdash (\overline{\tau})\ \mathtt{t} \leqslant (\overline{\tau'})\ \mathtt{p} \Rightarrow \theta_n}$$

$$\textsc{Inst-Construct-Exp-Left}\ \frac{let\ \tau = expand(\Gamma, \Phi, \mathtt{t}, \overline{\tau}) \qquad \Gamma, \Phi, \theta \vdash \tau \leqslant \tau' \Rightarrow \theta'}{\Gamma, \Phi, \theta \vdash (\overline{\tau})\ \mathtt{t} \leqslant \tau' \Rightarrow \theta'}$$

$$\textsc{Inst-Construct-Exp-Right}\ \frac{let\ \tau' = expand(\Gamma, \Phi, \mathtt{t}', \overline{\tau'}) \qquad \Gamma, \Phi, \theta \vdash \tau \leqslant \tau' \Rightarrow \theta'}{\Gamma, \Phi, \theta \vdash \tau \leqslant (\overline{\tau'})\ \mathtt{t}' \Rightarrow \theta'}$$

$$\textsc{Inst-Poly}\ \frac{let\ \theta' = \forall i.\theta \oplus [\alpha_i' \rightarrow \alpha_i] \qquad \Gamma, \Phi, \theta' \vdash \tau \leqslant \tau' \Rightarrow \theta''}{\Gamma, \Phi, \theta \vdash \forall \overline{\alpha}.\ \tau \leqslant \forall \overline{\alpha'}.\ \tau' \Rightarrow \theta''}$$

$$\textsc{Inst-Univar}\ \frac{[\alpha' \rightarrow \alpha] \in \theta}{\Gamma, \Phi, \theta \vdash \alpha \leqslant \alpha' \Rightarrow \theta}$$

$$\textsc{Inst-Rigid1}\ \frac{\Phi(\mathtt{t}) = \Phi(\tau)}{\Gamma.Types \oplus (\mathtt{t} : \mathcal{R}), \Phi, \theta \vdash \tau \leqslant \mathtt{t} \Rightarrow \theta}$$

$$\textsc{Inst-Rigid2}\ \frac{\Phi(\mathtt{t}) = \Phi(\tau)}{\Gamma.Types \oplus (\mathtt{t} : \mathcal{R}), \Phi, \theta \vdash \mathtt{t} \leqslant \tau \Rightarrow \theta}$$

$$\textsc{Inst-Variant}\ \frac{\begin{array}{c} T_{pres_i} .. T_{pres_j} \subseteq T'_{pres_i} .. T'_{pres_j} \qquad \Gamma, \Phi, \theta \vdash \rho \leqslant \rho' \Rightarrow \theta_\rho \\ \forall i.\ T_i \in \overline{T'} \implies \Gamma, \Phi, \theta_{i-1} \vdash \tau_{var_i} \leqslant \tau'_{var_i} \Rightarrow \theta_i \end{array}}{\Gamma, \Phi, \theta \vdash [(\rho)\ \overline{T\ of\ \tau_{var}} > T_{pres_i} .. T_{pres_j}] \leqslant [(\rho')\ \overline{T'\ of\ \tau'_{var}} > T'_{pres_i} .. T'_{pres_j}] \Rightarrow \theta_n}$$

$$\textsc{Inst-Nil}\ \Gamma, \Phi, \theta \vdash \epsilon \leqslant \epsilon \Rightarrow \theta$$

$$\textsc{Inst-Package}\ \frac{\begin{array}{c} let\ S = Sig(\mathtt{P}\ \mathbf{with}\ \overline{\mathbf{type}\ \mathtt{t} = \tau}) \\ let\ S' = Sig(\mathtt{P}'\ \mathbf{with}\ \overline{\mathbf{type}\ \mathtt{t}' = \tau'}) \qquad \Gamma, \Phi, \theta \vdash S' :> S \Rightarrow \theta' \end{array}}{\Gamma, \Phi, \theta \vdash (\mathbf{module}\ \mathtt{P}\ \mathbf{with}\ \overline{\mathbf{type}\ \mathtt{t} = \tau}) \leqslant (\mathbf{module}\ \mathtt{P}'\ \mathbf{with}\ \overline{\mathbf{type}\ \mathtt{t}' = \tau'}) \Rightarrow \theta'}$$

4 The type wellformedness condition

$$\textsc{Wf-Var}\ \Gamma, \Phi \vdash\ 'a\ \mathit{wf} \qquad \textsc{Wf-Fun}\ \frac{\Gamma, \Phi \vdash \tau_1\ \mathit{wf} \qquad \Gamma, \Phi \vdash \tau_2\ \mathit{wf}}{\Gamma, \Phi \vdash (l : \tau_1) \to \tau_2\ \mathit{wf}}$$

$$\textsc{Wf-Tuple}\ \frac{\forall \tau_i.\ \Gamma, \Phi \vdash \tau_i\ \mathit{wf}}{\Gamma, \Phi \vdash \tau_0 * .. * \tau_n\ \mathit{wf}}$$

$$\textsc{Wf-Construct}\ \frac{\begin{array}{c} \mathit{let}\ (\overline{\tau_{param}})\ \mathtt{t} = \Gamma.\mathit{Types}(\mathtt{t}) \\ \forall i. \Gamma, \Phi \vdash \tau_i\ \mathit{wf} \qquad \Gamma, \Phi, \Sigma \vdash (\overline{\tau})\ \mathtt{t} \leqslant (\overline{\tau_{param}})\ \mathtt{t} \Rightarrow \theta \end{array}}{\Gamma, \Phi \vdash (\overline{\tau})\ \mathtt{t}\ \ \mathit{wf}}$$

$$\textsc{Wf-Poly}\ \frac{\forall i.\ \Gamma, \Phi \vdash \tau_{\alpha_i} \prec \alpha_i \qquad \Gamma \oplus \overline{\alpha}, \Phi \vdash \tau\ \mathit{wf}}{\Gamma, \Phi \vdash \forall \overline{\tau_\alpha}.\ \tau\ \ \mathit{wf}} \qquad \textsc{Wf-Univar}\ \frac{\alpha \in \Gamma}{\Gamma, \Phi \vdash \alpha\ \mathit{wf}}$$

$$\textsc{Wf-Variant}\ \frac{\begin{array}{c} \forall i, j.\ i \neq j \implies T_i \neq T_j \\ \forall i. \Gamma, \Phi \vdash \tau_i\ \mathit{wf} \qquad \forall T_{pres_i} \in \{T_{pres_i}\ ..\ T_{pres_j}\}.\ T_{pres_i} \in \overline{T} \\ \Gamma, \Phi \vdash \rho \prec\ 'a \bigvee \Gamma, \Phi \vdash \rho \prec \epsilon \end{array}}{\Gamma, \Phi \vdash [(\rho)\ \overline{\sim T\ \mathtt{of}\ \tau} > T_{pres_i}\ ..\ T_{pres_j}]\ \mathit{wf}}$$

$$\textsc{Wf-Package}\ \frac{\begin{array}{c} \mathtt{P} \in \Gamma.\mathit{Modtypes} \qquad \forall i.\ \mathtt{t_i} \in \Gamma.\mathit{Modtypes}(\mathtt{P}) \\ \forall i. \Gamma, \Phi \vdash \tau_i\ \mathit{wf} \qquad \forall i\ \mathit{s.t.}\ \mathit{transparent}(\mathtt{P.t_i}).\ \Gamma, \Phi \vdash \tau_i \leqslant \mathtt{P.t_i} \end{array}}{\Gamma, \Phi \vdash (\mathtt{module}\ \mathtt{P}\ \mathtt{with}\ \overline{\mathit{type}\ \mathtt{t} = \tau})\ \mathit{wf}}$$

5 OCaml expressions (without objects)

$$\textsc{Tuple}\ \frac{\forall i.\ \Gamma,\Phi \vdash \langle e_i : \tau'_i \rangle \qquad \Gamma,\Phi \vdash \tau \prec \tau_0\ *\ ..\ *\ \tau_n \qquad \forall i. \Gamma,\Phi \vdash \tau_i \equiv \tau'_i}{\Gamma,\Phi \vdash \langle (e_0,\ ..,\ e_n) : \tau \rangle}$$

$$\textsc{Match}\ \frac{\begin{array}{c}\Gamma,\Phi \vdash \langle e : \tau_{scrut} \rangle \qquad \forall i.\ \Gamma,\Phi,\Sigma,\Sigma \vdash \langle p_i : \tau_i \rangle \Rightarrow (\overline{v_i : \tau_i}), (\overline{\tau_{\exists_i}}), \Phi_i \\ \forall i.\ let\ \Gamma_i = \Gamma \oplus_{\mathcal{V}} (\overline{v_i : \tau_i}) \oplus (\overline{\tau_{\exists_i}}) \\ \forall i.\ \Gamma_i,\Phi_i \vdash \langle e_i : \tau_{e_i} \rangle \qquad \forall i.\ \Gamma,\Phi \vdash \tau_{e_i}\ wf \qquad \forall i.\ \Gamma_i,\Phi_i \vdash \tau \equiv \tau_{e_i}\end{array}}{\Gamma,\Phi \vdash \langle \mathtt{match}\ e\ \mathtt{with}\ \overline{\mid p \rightarrow e} : \tau \rangle}$$

$$\textsc{Record}\ \frac{\begin{array}{c}let\ \tau_{rec} = \Gamma.Types(\tau) \qquad \forall i. let\ \tau_{l_i} = find\_label(\Gamma,\Phi,l_i,\tau_{rec}) \\ \forall i.\ \Gamma,\Phi \vdash \langle e : \tau_i \rangle \qquad \Gamma,\Phi,\Sigma \vdash \tau_0 \leqslant \tau_{l_0} \Rightarrow \theta_0 \\ \forall i_{\geqslant 1}.\ \Gamma,\Phi,\theta_{i-1} \vdash \tau_i \leqslant \tau_{l_i} \Rightarrow \theta_i \qquad let\ \tau_{inst} = \theta_n(\tau_{rec}) \qquad \Gamma,\Phi \vdash \tau_{inst} \equiv \tau\end{array}}{\Gamma,\Phi \vdash \langle \{l_0 = e_0;\ ..\ ; l_n = e_n\} : \tau \rangle}$$

$$\textsc{Field}\ \frac{\begin{array}{c}\Gamma,\Phi \vdash \langle e : \tau_e \rangle \qquad let\ \tau_{rec} = \Gamma.Types(\tau_e) \qquad let\ \tau_l = find\_label(\Gamma,\Phi,l,\tau_{rec}) \\ \Gamma,\Phi,\Sigma \vdash \tau \leqslant \tau_l \Rightarrow \theta \qquad let\ \tau_{inst} = \theta(\tau_{rec}) \qquad \Gamma,\Phi \vdash \tau_{inst} \equiv \tau_e\end{array}}{\Gamma,\Phi \vdash \langle e.l : \tau \rangle}$$

$$\textsc{Set-Field}\ \frac{\begin{array}{c}\Gamma,\Phi \vdash \langle e_1 : \tau_1 \rangle \qquad \Gamma,\Phi \vdash \langle e_2 : \tau_2 \rangle \\ let\ \tau_{rec} = \Gamma.Types(\tau_1) \qquad let\ \tau_l = find\_label(\Gamma,\Phi,l,\tau_{rec}) \\ label\_kind(\Gamma,\Phi,l,\tau_{rec})\ is\ Mutable \qquad \Gamma,\Phi,\Sigma \vdash \tau_2 \leqslant \tau_l \Rightarrow \theta \\ let\ \tau_{inst} = \theta(\tau_{rec}) \qquad \Gamma,\Phi \vdash \tau_{inst} \equiv \tau_1 \qquad \Gamma,\Phi \vdash \tau \equiv \mathtt{unit}\end{array}}{\Gamma,\Phi \vdash \langle e_1.l \leftarrow e_2 : \tau \rangle}$$

$$\textsc{Array}\ \frac{\begin{array}{c}\forall i.\ \Gamma,\Phi \vdash \langle e_i : \tau_i \rangle \\ \forall i_{\geqslant 1}.\ \Gamma,\Phi \vdash \tau_{i-1} \equiv \tau_i \qquad \Gamma,\Phi \vdash \tau \prec \tau_{arg}\ \mathtt{array} \qquad \Gamma,\Phi \vdash \tau_0 \equiv \tau_{arg}\end{array}}{\Gamma,\Phi \vdash \langle [|e_0;\ ..\ ; e_n|] : \tau \rangle}$$

$$\textsc{Sequence}\ \frac{\begin{array}{c}\Gamma,\Phi \vdash \langle e_1 : \tau_1 \rangle \\ \Gamma,\Phi \vdash \langle e_2 : \tau_2 \rangle \qquad \Gamma,\Phi \vdash \tau_1 \equiv \mathtt{unit} \qquad \Gamma,\Phi \vdash \tau \equiv \tau_2\end{array}}{\Gamma,\Phi \vdash \langle e_1;\ e_2 : \tau \rangle}$$

$$\textsc{While}\ \frac{\begin{array}{c}\Gamma,\Phi \vdash \langle e_1 : \tau_1 \rangle \qquad \Gamma,\Phi \vdash \langle e_2 : \tau_2 \rangle \\ \Gamma,\Phi \vdash \tau_1 \equiv \mathtt{bool} \qquad \Gamma,\Phi \vdash \tau_2 \equiv \mathtt{unit} \qquad \Gamma,\Phi \vdash \tau \equiv \mathtt{unit}\end{array}}{\Gamma,\Phi \vdash \langle \mathtt{while}\ e_1\ \mathtt{do}\ e_2\ \mathtt{done} : \tau \rangle}$$

$$\textsc{For}\ \frac{\begin{array}{c}\Gamma,\Phi \vdash \langle e_1 : \tau_1 \rangle \\ \Gamma,\Phi \vdash \langle e_2 : \tau_2 \rangle \qquad \Gamma,\Phi \vdash \tau_1 \equiv \mathtt{int} \qquad \Gamma,\Phi \vdash \tau_2 \equiv \mathtt{int} \\ \Gamma \oplus_{\mathcal{V}} (x,\tau_1), C.\Phi \vdash \langle e_3 : \tau_3 \rangle \qquad \Gamma,\Phi \vdash \tau_3 \equiv \mathtt{unit} \qquad \Gamma,\Phi \vdash \tau \equiv \mathtt{unit}\end{array}}{\Gamma,\Phi \vdash \langle \mathtt{for}\ x = e_1\ \mathtt{to}\ e_2\ \mathtt{do}\ e_3\ \mathtt{done} : \tau \rangle}$$

$$\textsc{Raise}\ \frac{\Gamma, \Phi \vdash \langle e : \tau_e \rangle \qquad \Gamma, \Phi \vdash (\tau_e \equiv \mathtt{exn}) \qquad \Gamma, \Phi \vdash (\tau \equiv \mathtt{unit})}{\Gamma, \Phi \vdash \langle \mathbf{assert}\ e : \tau \rangle}$$

$$\textsc{Assert}\ \frac{\Gamma, \Phi \vdash \langle e : \tau_e \rangle \qquad \Gamma, \Phi \vdash (\tau_e \equiv \mathtt{bool}) \qquad \Gamma, \Phi \vdash (\tau \equiv \mathtt{unit})}{\Gamma, \Phi \vdash \langle \mathbf{assert}\ e : \tau \rangle}$$

$$\textsc{Assert-False}\ \frac{\Gamma, \Phi \vdash \langle \mathit{false} : \tau_e \rangle \qquad \Gamma, \Phi \vdash (\tau_e \equiv \mathtt{bool})}{\Gamma, \Phi \vdash \langle \mathbf{assert}\ \mathit{false} : \tau \rangle}$$

$$\textsc{Lazy}\ \frac{\Gamma, \Phi \vdash \langle e : \tau_e \rangle \qquad \Gamma, \Phi \vdash \tau \prec \tau_{arg}\ \mathtt{Lazy.t} \qquad \Gamma, \Phi \vdash \tau_e \equiv \tau_{arg}}{\Gamma, \Phi \vdash \langle \mathbf{lazy}\ e : \tau \rangle}$$

$$\textsc{Variant-Const}\ \frac{\Gamma, \Phi \vdash \tau \prec [(\rho)\ ..\ T\ ..\ >\ ..\ T\ ..\ ]}{\Gamma, \Phi \vdash \langle `T : \tau \rangle}$$

$$\textsc{Variant}\ \frac{\begin{array}{c}\Gamma, \Phi \vdash \tau \prec [(\rho) ..\ T\ \mathit{of}\ \tau_{arg}\ ..\ >\ ..\ T\ ..\ ] \\ \Gamma, \Phi \vdash \langle e : \tau_e \rangle \qquad \Gamma, \Phi \vdash \tau_{arg} \equiv \tau_e\end{array}}{\Gamma, \Phi \vdash \langle `T\ e : \tau \rangle}$$

$$\textsc{Constraint}\ \frac{\begin{array}{c}\Gamma, \Phi, \Sigma \vdash \langle t : \tau_{cstr} \rangle \Rightarrow \mathcal{V} \\ \Gamma, \Phi \vdash \langle e : \tau_e \rangle \qquad \Gamma, \Phi, \Sigma \vdash \tau_e \leqslant \tau_{cstr} \Rightarrow \theta \qquad \Gamma, \Phi \vdash \tau \equiv \tau_e\end{array}}{\Gamma, \Phi \vdash \langle (e : t) : \tau \rangle}$$

$$\textsc{Newtype}\ \frac{\begin{array}{c}\mathit{let}\ \Gamma' = \Gamma.\mathit{Types} \oplus (\mathtt{t} : \mathcal{R}) \qquad \Gamma' \vdash \langle e : \tau_e \rangle \qquad \mathit{let}\ \alpha\ \mathit{fresh}(\Gamma) \\ \mathit{let}\ \theta = [\mathtt{t} \to \alpha] \qquad \mathit{let}\ \tau_t = \theta(\tau_e) \qquad \Gamma, \Phi \vdash \tau \equiv \tau_t \qquad \Gamma, \Phi \vdash \tau\ \mathit{wf}\end{array}}{\Gamma, \Phi \vdash \langle \mathbf{fun}\ (\mathbf{type}\ \mathtt{t}) \to e : \tau \rangle}$$

$$\textsc{Let-Module}\ \frac{\begin{array}{c}\Gamma, \Phi \vdash \langle M : \mathcal{M} \rangle \\ \Gamma.\mathit{Modules} \oplus (\mathtt{I}, \mathcal{M}), \Phi \vdash \langle e : \tau_e \rangle \qquad \Gamma, \Phi \vdash \tau_e\ \mathit{wf} \qquad \Gamma, \Phi \vdash \tau \equiv \tau_e\end{array}}{\Gamma, \Phi \vdash \langle \mathbf{let\ module}\ \mathtt{I} = M\ \mathbf{in}\ e : \tau \rangle}$$

6 OCaml patterns

$$\textsc{Pat-Record}\ \frac{\begin{array}{c} let\ \tau_{rec} = find\_record(\Gamma,\Phi,\tau) \qquad \forall i.\ let\ \tau_{l_i} = find\_label(\Gamma,\Phi,l_i,\tau_{rec}) \\ \Gamma,\Phi,\mathcal{V},\mathcal{T}\ \vdash \langle p_0 : \tau_0\rangle \Rightarrow \mathcal{V}_0,\mathcal{T}_0,\Phi_0 \\ \forall i_{\geqslant 1}.\ \Gamma,\Phi_{i-1},\mathcal{V}_{i-1},\mathcal{T}_{i-1}\ \vdash \langle p_i : \tau_i\rangle \Rightarrow \mathcal{V}_i,\mathcal{T}_i,\Phi_i \\ \Gamma,\Phi_0 \vdash \tau_0 \leqslant \tau_{l_0} \Rightarrow \theta_0 \qquad \forall i_{\geqslant 1}.\ \Gamma,\Phi_i,\theta_{i-1} \vdash \tau_i \leqslant \tau_{l_i} \Rightarrow \theta_i \\ let\ \tau_{inst} = \theta_n(\tau_{rec}) \qquad \Gamma,\Phi_n \vdash \tau_{inst} \equiv \tau \end{array}}{\Gamma,\Phi,\mathcal{V},\mathcal{T}\ \vdash \langle \{l_0 = p_0;\ ..\ ; l_n = p_n\} : \tau\rangle \Rightarrow \mathcal{V}_n,\mathcal{T}_n,\Phi_n}$$

$$\textsc{Pat-Array}\ \frac{\begin{array}{c} \Gamma,\Phi,\mathcal{V},\mathcal{T}\ \vdash \langle p_0 : \tau_0\rangle \Rightarrow \mathcal{V}_0,\mathcal{T}_0,\Phi_0 \\ \forall i_{\geqslant 1}.\ \Gamma,\Phi_{i-1},\mathcal{V}_{i-1},\mathcal{T}_{i-1} \vdash \langle p_i : \tau_i\rangle \Rightarrow \mathcal{V}_i,\mathcal{T}_i,\Phi_i \\ \forall i_{\geqslant 1}.\ \Gamma,\Phi_i \vdash \tau_{i-1} \equiv \tau_i \qquad \Gamma,\Phi \vdash \tau \prec \tau'\ \texttt{array} \qquad \Gamma,\Phi_0 \vdash \tau_0 \equiv \tau \end{array}}{\Gamma,\Phi,\mathcal{V},\mathcal{T} \vdash \langle [|p_0;\ ..\ ; p_n|] : \tau\rangle \Rightarrow \mathcal{V}_n,\mathcal{T}_n,\Phi_n}$$

$$\textsc{Pat-Lazy}\ \frac{\begin{array}{c} \Gamma,\Phi,\mathcal{V},\mathcal{T} \vdash p : \tau_p \Rightarrow \mathcal{V}',\mathcal{T}',\Phi' \\ \Gamma,\Phi \vdash \tau \prec \tau'\ \texttt{Lazy.t} \qquad \Gamma,\Phi' \vdash \tau' \equiv \tau_p \end{array}}{\Gamma,\Phi,\mathcal{V} \vdash \langle \textbf{lazy}\ p : \tau\rangle \Rightarrow \mathcal{V}',\mathcal{T}',\Phi'}$$

$$\textsc{Pat-Variant-Const}\ \frac{\Gamma,\Phi_p \vdash \tau \prec [(\rho)\ T > ..\ ]}{\Gamma,\Phi,\mathcal{V},\mathcal{T} \vdash \langle `T : \tau\rangle \Rightarrow \mathcal{V},\mathcal{T},\Phi}$$

$$\textsc{Pat-Variant}\ \frac{\begin{array}{c} \Gamma,\Phi \vdash \tau \prec [(\rho)\ T\ of\ \tau_{arg} > ..\ ] \\ \Gamma,\Phi,\mathcal{V},\mathcal{T} \vdash \langle p : \tau_p\rangle \Rightarrow \mathcal{V}',\mathcal{T}',\Phi' \qquad \Gamma,\Phi' \vdash \tau_{arg} \equiv \tau_p \end{array}}{\Gamma,\Phi,\mathcal{V},\mathcal{T} \vdash \langle `T\ p : \tau\rangle \Rightarrow \mathcal{V}',\mathcal{T}',\Phi'}$$

7 OCaml type annotations

These annotations can appear as constraints on expressions, as types of labels from record declaration or arguments of constructor declarations.

$$\textsc{Core-Var-Bound}\ \frac{\Gamma, \Phi \vdash \tau \equiv \tau_a}{\Gamma, \Phi, \mathcal{V} \oplus ('a, \tau_a) \vdash \langle 'a : \tau \rangle \Rightarrow \mathcal{V} \oplus ('a, \tau_a)}$$

$$\textsc{Core-Var-Unbound}\ \frac{}{\Gamma, \Phi, \mathcal{V} \vdash \langle 'a : \tau \rangle \Rightarrow V \oplus ('a, \tau)}$$

$$\textsc{Core-Any}\ \Gamma, \Phi, \mathcal{V} \vdash \langle \_ : \tau) \Rightarrow \mathcal{V}$$

$$\textsc{Core-Arrow}\ \frac{\begin{array}{c}\Gamma, \Phi, \mathcal{V} \vdash \langle t_1 : \tau_1 \rangle \Rightarrow \mathcal{V}_1 \qquad \Gamma, \Phi, \mathcal{V}_1 \vdash \langle t_2 : \tau_2' \rangle \Rightarrow \mathcal{V}_2 \\ l = l' \qquad \Gamma, \Phi \vdash \tau_d \equiv \tau_1 \qquad \Gamma, \Phi \vdash \tau_{cd} \equiv \tau_2\end{array}}{\Gamma, \Phi, \mathcal{V} \vdash \langle (l : t_1) \to t_2 :\ (l' : \tau_d) \to \tau_{cd} \rangle \Rightarrow \mathcal{V}_2}$$

$$\textsc{Core-Tuple}\ \frac{\begin{array}{c}\Gamma, \Phi, \mathcal{V} \vdash \langle t_0 : \tau_0' \rangle \Rightarrow \mathcal{V}_0 \\ \forall i_{\geqslant 1}.\ \Gamma, \Phi, \mathcal{V}_{i-1} \vdash \langle t_i : \tau_i' \rangle \Rightarrow \mathcal{V}_i \qquad \forall i.\ \Gamma, \Phi \vdash \tau_i \equiv \tau_i'\end{array}}{\Gamma, \Phi, \mathcal{V} \vdash \langle t_0 * .. * t_n : \tau_0 * .. * \tau_n \rangle \Rightarrow \mathcal{V}_n}$$

$$\textsc{Core-Constr}\ \frac{\begin{array}{c}\Gamma, \Phi, \mathcal{V} \vdash \langle t_0 : \tau_0' \rangle \Rightarrow \mathcal{V}_0 \qquad \forall i_{\geqslant 1}.\ \Gamma, \Phi, \mathcal{V}_{i-1}, \vdash \langle t_i : \tau_i' \rangle \Rightarrow \mathcal{V}_i \\ \forall i.\ \Gamma, \Phi \vdash \tau_i \equiv \tau_i' \qquad \mathtt{t} = \mathtt{t}' \qquad \Gamma, \Phi \vdash (\overline{\tau})\ \mathtt{t}'\ wf\end{array}}{\Gamma, \Phi, \mathcal{V} \vdash \langle (\overline{t})\ \mathtt{t} : (\overline{\tau})\ \mathtt{t}' \rangle \Rightarrow \mathcal{V}_n}$$

$$\textsc{Core-Poly}\ \frac{\begin{array}{c}\forall i.\ \Gamma, \Phi \vdash \tau_i \prec \alpha_i \\ \Gamma, \Phi, \mathcal{V} \oplus (\overline{'a, \alpha}) \vdash \langle t : \tau_{poly} \rangle \Rightarrow \mathcal{V}_{poly} \qquad \Gamma, \Phi \vdash \tau \equiv \tau_{poly}\end{array}}{\Gamma, \Phi, \mathcal{V} \vdash \langle \overline{'a}.\ t : \overline{\tau}.\ \tau \rangle \Rightarrow \mathcal{V}_{poly}}$$

$$\textsc{Core-Alias-Nonrec}\ \frac{\Gamma, \Phi, \mathcal{V} \vdash \langle t : \tau_{al} \rangle \Rightarrow \mathcal{V}_{al} \qquad 'a \notin \mathcal{V}_{al} \qquad \Gamma, \Phi \vdash \tau \equiv \tau_{al}}{\Gamma, \Phi, \mathcal{V} \vdash \langle t\ \mathbf{as}\ 'a \rangle : \tau \Rightarrow \mathcal{V}_{al}}$$

$$\textsc{Core-Variant-Static}\ \frac{\begin{array}{c}\Gamma, \Phi, \mathcal{V} \vdash \langle t_0 : \tau_0' \rangle \Rightarrow \mathcal{V}_0 \qquad \forall i.\ \Gamma, \Phi, \mathcal{V}_{i-1} \vdash \langle t_i : \tau_i' \rangle \Rightarrow \mathcal{V}_i \\ \forall i.\ T_i \in [T_{pres_i}\ ..\ T_{pres_j}] \qquad \forall i.\ \Gamma, \Phi \vdash \tau_i \equiv \tau_i' \qquad \Gamma, \Phi \vdash \rho \prec \epsilon\end{array}}{\Gamma, \Phi, \mathcal{V} \vdash \langle [\overline{T\ of\ t}] : [(\rho)\ \overline{\mid T\ ?of\ \tau} > T_{pres_i}..T_{pres_j}] \rangle \Rightarrow \mathcal{V}_n}$$

$$\textsc{Core-Variant-Closed}\ \frac{\begin{array}{c}\Gamma, \Phi, \mathcal{V} \vdash \langle t_0 : \tau_0' \rangle \Rightarrow \mathcal{V}_0 \\ \forall i.\ \Gamma, \Phi, \mathcal{V}_{i-1} \vdash \langle t_i : \tau_i' \rangle \Rightarrow \mathcal{V}_i \qquad \forall i.\ T_{pres_i} \in [T'_{pres_i}\ ..\ T'_{pres_j}] \\ \forall i.\ \Gamma, \Phi \vdash \tau_i \equiv \tau_i' \qquad \Gamma, \Phi \vdash \rho \prec' a\end{array}}{\Gamma, \Phi, \mathcal{V} \vdash \langle [\overline{T\ of\ t} > T_{pres_i}\ ..\ T_{pres_j}] : [(\rho)\ \overline{\mid T\ ?of\ \tau} > T_{pres_i}\ ..\ T_{pres_j}] \rangle \Rightarrow \mathcal{V}_n}$$

$$\textsc{Core-Variant-Open}\ \frac{\begin{array}{c}\Gamma, \Phi, \mathcal{V} \vdash \langle t_0 : \tau_0' \rangle \Rightarrow \mathcal{V}_0 \qquad \forall i.\ \Gamma, \Phi, \mathcal{V}_{i-1} \vdash \langle t_i : \tau_i' \rangle \Rightarrow \mathcal{V}_i \\ \forall i.\ T_i \in [T_{pres_i}\ ..\ T_{pres_j}] \qquad \forall i.\ \Gamma, \Phi \vdash \tau_i \equiv \tau_i' \qquad \Gamma, \Phi \vdash \rho \prec' a\end{array}}{\Gamma, \Phi, \mathcal{V} \vdash \langle [> \overline{T\ of\ t}] : [(\rho)\ \overline{\mid T\ ?of\ \tau} > T_{pres_i}\ ..\ T_{pres_j}] \rangle \Rightarrow \mathcal{V}_n}$$

$$\textsc{Core-Package}\ \frac{\begin{array}{c}\mathtt{S} = \mathtt{S}' \qquad \forall i.\ \mathtt{t_i} = \mathtt{t}'_\mathtt{i} \\ \Gamma, \Phi, \mathcal{V} \vdash \langle t_0 : \tau_0' \rangle \Rightarrow \mathcal{V}_0 \qquad \forall i_{\geqslant 1}.\ \Gamma, \Phi, \mathcal{V}_{i-1} \vdash \langle t_i : \tau_i' \rangle \Rightarrow \mathcal{V}_i \\ \forall i.\ \Gamma, \Phi \vdash \tau_i \equiv \tau_i' \qquad \Gamma, \Phi \vdash (\mathbf{module}\ S'\ \mathbf{with\ type}\ \overline{\mathtt{t}' = \tau}))\ wf\end{array}}{\Gamma, \Phi, \mathcal{V} \vdash \langle (\mathbf{module}\ \mathtt{S}\ \mathbf{with\ type}\ \overline{\mathtt{t} = t}) : (\mathbf{module}\ \mathtt{S}'\ \mathbf{with\ type}\ \overline{\mathtt{t}' = \tau}) \rangle \Rightarrow \mathcal{V}_n}$$