Elaboration Notes

These notes were distilled and typeset by Jacob Potter `<jdpotter@andrew.cmu.edu>`, based on:

- Lectures and theory by Karl Crary `<crary@cs.cmu.edu>`
- My own lecture notes and code
- Lecture notes taken by Ben Segall `<bsegall@cs.cmu.edu>`

Some general formatting and implementation notes:

- "$\Gamma \vdash \tau : \mathrm{T}$" in a premise means that τ might have a free α in it, and so it must have inhabitant (fstsg σ) substituted for α.
- "$\Gamma, \alpha : \sigma$" means "`extend G a (fstsg sg)`".
- "$\Gamma \vdash \mathcal{E} \rhd \mathit{longid}$ in ..." is `Resolve.resolve`.
- "$\Gamma \vdash \mathcal{E} \rhd \mathit{ty} \leadsto \tau$" is `ElaborateType.elabTp`.
- "$\Gamma \vdash \tau \equiv \ldots : \mathrm{T}$" means to pattern match on the output of `whnf` τ.
- The type that is called `unit` in ML is always written as $\times[]$.

1 elabMatch with v_{fail}

[(PAT, EXP)]

$$\frac{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd \text{pat on } v : \tau_1 \text{ with } v_{\text{fail}} \leadsto M : \sigma \\ \Gamma, \alpha : \sigma \vdash \mathcal{E}@(\alpha/m : \sigma) \rhd \text{exp} \leadsto e : \tau_2 \qquad \Gamma \vdash \tau_2 : \mathrm{T}\end{array}}{\Gamma \vdash \mathcal{E} \rhd \text{pat} \Rightarrow \text{exp} \ \text{ on } v : \tau_1 \text{ with } v_{\text{fail}} \leadsto \text{let } \alpha/m = M \text{ in } (e : \tau_2) : \tau_2}$$

(PAT, EXP) :: MATCH

$$\frac{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd \text{pat on } v : \tau_1 \text{ with } v_{\text{fail}} \leadsto M : \sigma \\ \Gamma, \alpha : \sigma \vdash \mathcal{E}@(\alpha/m : \sigma) \rhd \text{exp} \leadsto e : \tau_2 \\ \Gamma \vdash \tau_2 : \mathrm{T} \qquad \Gamma \vdash \mathcal{E} \rhd \text{match on } v : \tau_1 \text{ with } v_{\text{fail}} \leadsto e' : \tau_2\end{array}}{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd \text{pat} \Rightarrow \text{exp} \mid \text{match on } v : \tau_1 \text{ with } v_{\text{fail}} \leadsto \\ \text{handle}(\text{let } \alpha/m = M \text{ in } (e : \tau_2), x, \text{iftag}(v_{\text{fail}}, x,, e', \text{raise}_{\tau_2}\, x)) : \tau_2\end{array}}$$

2 elabMatch

$$\frac{\Gamma \vdash \mathcal{E} \rhd \text{match on } v : \tau_1 \text{ with } v_{\text{fail}} \leadsto e : \tau}{\Gamma \vdash \mathcal{E} \rhd \text{match on } v : \tau_1 \leadsto (\text{let } v_{\text{fail}} = \text{newtag}_{\times[]} \text{ in } e) : \tau}$$

3 elabPattern

PIDENT

$$\frac{}{\Gamma \vdash \mathcal{E} \rhd \mathrm{id} \text{ on } e : \tau \text{ with } v_{\mathrm{fail}} \rightsquigarrow (\mathrm{in}_{\mathrm{VAL\ id}} \lhd \mathrm{id} \rhd) : (\mathrm{VAL\ id} : \lhd \tau \rhd)}$$

PINT

$$\frac{}{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd i \text{ on } e : \tau \text{ with } v_{\mathrm{fail}} \rightsquigarrow \\ \left(\mathrm{let}\ \_/\_ = \lhd \mathrm{if}\ e = i \text{ then } \langle\rangle \text{ else } \mathrm{raise}_{\times[]}\left(\mathrm{tag}_{v_{\mathrm{fail}}}\langle\rangle\right) \rhd \text{ in } * : 1\right) : 1\end{array}}$$

Note: In the following rule, $\langle M1, M2\rangle$ is shorthand for $\langle \_/\_ = M1, M2\rangle$ and $\sigma_1 \times \sigma_2$ is shorthand for $\Sigma_{\_} : \sigma_1 \times \sigma_2$.

PTUPLE

$$\frac{\begin{array}{c}\Gamma \vdash \tau \equiv \times[\tau_0, \ldots \tau_{n-1}] : \mathrm{T} \\ \Gamma \vdash \mathcal{E} \rhd pat_i \text{ on } \pi_i e : \tau_i \text{ with } v_{\mathrm{fail}} \rightsquigarrow M_i : \sigma_i \quad (\text{for } i \in [0, \ldots, n-1])\end{array}}{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd (pat_0, \ldots pat_n) \text{ on } e : \tau \text{ with } v_{\mathrm{fail}} \rightsquigarrow \\ \langle M_0, \langle M_1, \langle \ldots \langle M_{n-1}, *\rangle\rangle\rangle : \sigma_0 \times (\sigma_1 \times (\ldots(\sigma_{n-1} \times 1)))\end{array}}$$

PAPP

$$\frac{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd longid \text{ in VAL} \rightsquigarrow M_0 : (\mathrm{DCON} : \lhd \tau_0 \rhd) \\ \Gamma \vdash \tau_0 \equiv \tau_c \times (\tau \to (\times[] + \tau')) : \mathrm{T} \\ \Gamma, \alpha : (\mathrm{checkCon}\ \Gamma\ \tau') \vdash \mathcal{E}@(\alpha/m : \lhd \tau' \rhd) \rhd pat \text{ on } \mathrm{Snd}(m) : \tau' \text{ with } v_{\mathrm{fail}} \rightsquigarrow \sigma : M\end{array}}{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd longid\ pat \text{ on } e : \tau \text{ with } v_{\mathrm{fail}} \rightsquigarrow \\ \left(\mathrm{let}\ \alpha/m = \lhd \left(\begin{array}{c}\mathrm{case}\ (\pi_1(\mathrm{Snd}(\mathrm{out}\ M_0)))\ e \text{ of } \mathrm{inj}_0\,\_ \Rightarrow \mathrm{raise}_{\tau'}(\mathrm{tag}(v_{\mathrm{fail}}, \langle\rangle)) \\ \mid \mathrm{inj}_1 x \Rightarrow x\end{array}\right) \rhd \text{ in } M\right) : \sigma\end{array}}$$

4 elabTerm

`Tint` and `Tstring` should be obvious. `Tprim` is handled with `ElaboratePrim.elabPrim`. Note that there are two rules for `Tvar` depending on the result of resolve.

TVAR : VALUE

$$\frac{\Gamma \vdash \mathcal{E} \rhd longid \text{ in VAL} \rightsquigarrow M : \lhd \tau \rhd}{\Gamma \vdash \mathcal{E} \rhd longid \rightsquigarrow \mathrm{Snd}(M) : \tau}$$

TVAR : DATA CONSTRUCTOR

$$\frac{\Gamma \vdash \mathcal{E} \rhd longid \text{ in VAL} \rightsquigarrow M : (\mathrm{DCON} : \lhd \tau \rhd) \qquad \Gamma \vdash \tau \equiv \tau_c \times \tau_d : \mathrm{T}}{\Gamma \vdash \mathcal{E} \rhd longid \rightsquigarrow \pi_0(\mathrm{Snd}(\mathrm{out}\ M)) : \tau_c}$$

TAPP

$$\frac{\Gamma \vdash \mathcal{E} \rhd exp_1 \rightsquigarrow e_1 : \tau_1 \to \tau_2 \qquad \Gamma \vdash \mathcal{E} \rhd exp_2 \rightsquigarrow e_2 : \tau_1}{\Gamma \vdash \mathcal{E} \rhd exp_1\ exp_2 \rightsquigarrow e_1 e_2 : \tau_2}$$

TTUPLE
$$\frac{\Gamma \vdash \mathcal{E} \rhd exp_1 \rightsquigarrow e_1 : \tau_1 \ldots \Gamma \vdash \mathcal{E} \rhd exp_n \rightsquigarrow e_n : \tau_n}{\Gamma \vdash \mathcal{E} \rhd \langle exp_1, \ldots, exp_n \rangle \rightsquigarrow \langle e_1, \ldots, e_n \rangle : \times[\tau_1, \ldots, \tau_n]}$$

TLET
$$\frac{\Gamma \vdash \mathcal{E} \rhd decls \rightsquigarrow M : \sigma \qquad \Gamma, \alpha : \sigma \vdash E@(\alpha/m : \sigma) \rhd exp \rightsquigarrow e : \tau \qquad \Gamma \vdash \tau : \mathrm{T}}{\Gamma \vdash \mathcal{E} \rhd \text{let } decls \text{ in } exp \rightsquigarrow (\text{let } \alpha/m = M \text{ in } e : \tau) : \tau}$$

5 elabDecl

Ddata is handled with `ElaborateDatatype.elabDatatype`.

DVAL
$$\frac{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd exp \rightsquigarrow e : \tau \\ \Gamma, \alpha_{\text{fail}} : \mathrm{T} \rhd \mathcal{E}@(a/m : \lhd \tau \rhd)@(\alpha_{\text{fail}}/m_{\text{fail}} : \lhd \text{tag}_{\times[]} \rhd) \rhd pat \text{ on } \mathrm{Snd}(m) \text{ with } \mathrm{Snd}(m_{\text{fail}}) \rightsquigarrow M : \sigma\end{array}}{\Gamma \vdash \mathcal{E} \rhd \text{val } pat = exp \rightsquigarrow (\text{let } \alpha_{\text{fail}}/m_{\text{fail}} = \lhd \text{newtag } \langle\rangle \rhd \text{ in let } \alpha/m = \lhd e \rhd \text{ in } M) : \sigma}$$

DTYPE
$$\frac{\Gamma \vdash \mathcal{E} \rhd ty \rightsquigarrow \tau}{\Gamma \vdash \mathcal{E} \rhd \text{type } id = ty \rightsquigarrow \text{in}_{\text{CON } id} (\!| \tau |\!) : ((\text{CON } id) : (\!| S(\tau) |\!))}$$

DOPEN
$$\frac{\Gamma \vdash \mathcal{E} \rhd longid \text{ in MOD} \rightsquigarrow M : \sigma}{\Gamma \vdash \mathcal{E} \rhd \text{open } id \rightsquigarrow M : \sigma}$$

DLOCAL
$$\frac{\Gamma \vdash \mathcal{E} \rhd dec_1 \rightsquigarrow M_1 : \sigma_1 \qquad \Gamma, \alpha : \sigma_1 \vdash \mathcal{E}@(\alpha/(\text{out } m) : \sigma_1) \rhd dec_2 \rightsquigarrow M_2 : \sigma_2}{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd \text{local } dec_1 \text{ in } dec_2 \text{ end} \rightsquigarrow \\ \langle \alpha/m = \text{in}_{\text{HIDE}} M_1, M_2 \rangle : \exists \alpha : (\text{HIDE} : \sigma_1).\sigma_2\end{array}}$$

DMODULE
$$\frac{\Gamma \vdash \mathcal{E} \rhd mod \rightsquigarrow M : \sigma}{\Gamma \vdash \mathcal{E} \rhd \text{structure } id = mod \rightsquigarrow \text{in}_{\text{MOD } id} M : (\text{MOD } id : \sigma)}$$

Dfun first requires some definitions:

$$\varphi_\tau = \mu\alpha.\alpha \to ((\times[] \to \tau) \to \tau) \to \tau$$
$$\theta_\tau = \text{roll}_{\varphi_\tau} (\lambda z : \varphi_\tau.\ \lambda f : (\times[] \to \tau) \to \tau.\ f\,(\lambda\_ : \times[].\ ((\text{unroll } z)z)f))$$
$$\Theta_\tau = (\text{unroll } \theta_\tau)\theta_\tau$$
$$\text{fix}_\tau x.\ e = (\lambda\_ : \times[].\ \Theta_\tau(\lambda x' : \times[] \to \tau.\ \text{let } x = x' \text{ in } e))\,\langle\rangle$$

DFUN
$$\frac{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd ty_1 \rightsquigarrow \tau_1 \qquad \Gamma \vdash \mathcal{E} \rhd ty_2 \rightsquigarrow \tau_2 \\ \Gamma, a : (\Sigma\_ : (\text{VAL } id_1 : \lhd \tau_1 \to \tau_2 \rhd).(\text{VAL } id_2 : \tau_1)) \vdash \mathcal{E}@(a/m : \text{same Ssigma as added to } \Gamma) \rhd \text{exp} : \tau_1\end{array}}{\begin{array}{c}\Gamma \vdash \mathcal{E} \rhd \text{fun} id_1(id_2 : ty_1) : ty_2 = \text{exp} \rightsquigarrow \\ \text{in}_{\text{VAL } id_1} \lhd \text{fix}_{\tau_1 \to \tau_2} f.\,(\lambda x : \tau_1.\text{let } \alpha/m = \langle \_/\_ = \text{in}_{\text{VAL } id_1} \lhd f\langle\rangle \rhd, \text{in}_{\text{VAL } id_2} \lhd x \rhd \rangle \text{ in } e : \tau_2) \rhd \\ : (\text{VAL } id_1 : \lhd \tau_1 \to \tau_2 \rhd)\end{array}}$$

6 elabDecls

$$\frac{}{\Gamma \vdash \mathcal{E} \rhd \epsilon \rightsquigarrow * : 1}\ \textsc{nil}$$

$$\frac{\Gamma \vdash \mathcal{E} \rhd dec_1 \rightsquigarrow M_1 : \sigma_1 \qquad \Gamma, \alpha : \sigma 1 \vdash \mathcal{E}@(\alpha/m : \sigma_1) \rhd dec_2 \rightsquigarrow M_2 : \sigma_2}{\Gamma \vdash \mathcal{E} \rhd dec_1\ dec_2 \rightsquigarrow \langle \alpha/m = M_1, M_2 \rangle : \Sigma\alpha : \sigma_1 . \sigma_2}\ \textsc{decl :: decls}$$

7 elabModule

The code for `Mseal` is written for us. You get a module, opacity, and signature; call `elabModule` and `elabSg`, and then pass the results (and the opacity) to `Ascribe.ascribe`.

$$\frac{\Gamma \vdash \mathcal{E} \rhd longid \text{ in MOD} \rightsquigarrow M : \sigma}{\Gamma \vdash \mathcal{E} \rhd longid \rightsquigarrow M : \sigma}\ \textsc{Mident}$$

$$\frac{\Gamma \vdash \mathcal{E} \rhd dec \rightsquigarrow M : \sigma}{\Gamma \vdash \mathcal{E} \rhd \text{struct } dec \text{ end} \rightsquigarrow M : \sigma}\ \textsc{Mstruct}$$