Abstract Binding Trees

Jon Sterling and Darin Morrison

1 Preliminaries

Fix a set $\mathscr{S}$ of *sorts*. We will say s *sort* when $s \in S$. A valence $\{\vec{p}\}[\vec{q}].s$ specifies an expression of sort s which binds symbols in $\vec{p}$ and variables in $\vec{q}$.

$$\frac{s\ \mathit{sort} \quad p_i\ \mathit{sort}\ \ (i \leq m) \quad q_i\ \mathit{sort}\ \ (i \leq n)}{\{p_0, \dots, p_m\}[q_0, \dots, q_n].s\ \mathit{valence}}$$

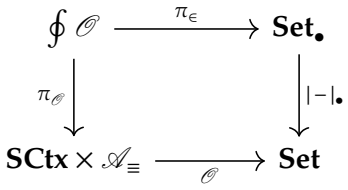
An arity $(\vec{v})s$ specifies an operator of sort s with arguments of valences $\vec{v}$. We will call the set of valences $\mathscr{V}$, and the set of arities $\mathscr{A}$.

$$\frac{s\ \mathit{sort} \quad v_i\ \mathit{valence}\ \ (i \leq n)}{(v_0, \dots, v_n)s\ \mathit{arity}}$$

Let $\mathbb{I}$ be an infinite set of symbols. Let $\mathbb{F}$ be the category of finite subsets of $\mathbb{I}$ and their injective maps; then the comma construction $\mathbf{SCtx} \triangleq \mathbb{F} \downarrow \mathscr{S}_{\equiv}$, with $\mathscr{S}_{\equiv}$ the discrete category on the set $\mathscr{S}$, is the category of contexts of symbols, whose objects are finite sets of symbols U and sort-assignments $\mathfrak{s} : U \to \mathscr{S}$, and whose morphisms are sort-preserving renamings; we will write Υ for a symbol context $(U, \mathfrak{s})$.

Then, fix a covariant presheaf (copresheaf) of operators $\mathscr{O} : \mathbf{SCtx} \times \mathscr{A}_{\equiv} \to \mathbf{Set}$ such that the arrows in $\mathbf{SCtx}$ lift to renamings of operators' parameters. Via the Grothendieck construction[1] $\oint(-) : \mathbf{Set}^{C} \to \mathbf{Cat}$ on operators, we have a category of objects $\langle\langle \Upsilon, a\rangle, \vartheta\rangle \in \oint \mathscr{O}$ for $\vartheta \in \mathscr{O}\langle \Upsilon, a\rangle$ and morphisms $\oint \mathscr{O}\,[\langle\langle \Upsilon, a\rangle, \vartheta\rangle, \langle\langle \Upsilon', a'\rangle, \vartheta'\rangle]$ for $\langle \varrho, f\rangle : \langle \Upsilon, a\rangle \to \langle \Upsilon', a'\rangle$ such that $\mathscr{O}\langle \varrho, f\rangle(\vartheta) = \vartheta' \in \mathscr{O}\langle \Upsilon', a'\rangle$. Equivalently, $\oint \mathscr{O}$ is the pullback of $\mathscr{O}$ along the universal $\mathbf{Set}$-bundle where $\pi_{\mathscr{O}}$ is a discrete Grothendieck opfibration and $|-|_\bullet$ is the forgetful functor from pointed sets:

[1] In this case, $C \oint \Psi$ represents the category of elements of a copresheaf $\Psi : C \to \mathbf{Set}$ but we keep the C implicit and simply refer to it as the Grothendieck construction. Alternatively, this construction can be understood as a coend $C \oint \Psi \cong \int^{c \in C} c/C \otimes \Psi(c)_{\equiv}$.



$$\frac{\vartheta \in \mathscr{O}\langle \Upsilon, a\rangle}{\Upsilon \Vdash \vartheta : a}$$

The judgment $\Upsilon \Vdash \vartheta : a$ enjoys the structural properties of weakening and exchange via the functoriality of $\mathscr{O}$.

Examples Operators are defined by specifying the fibers of $\pi_{\mathscr{O}}$ in which they reside. For instance, consider the lambda calculus with a single sort, exp; we give its signature by asserting the following about its operators:

$$\Upsilon \Vdash \lambda : (\mathsf{[exp].\,exp})\mathsf{exp}$$
$$\Upsilon \Vdash \mathsf{ap} : (\mathsf{.\,exp}, \mathsf{.\,exp})\mathsf{exp}$$

So far, we have made no use of symbols and parameters; however, consider the extension of the calculus with assignables (references):

$$\Upsilon \Vdash \mathsf{decl} : (\mathsf{.\,exp}, \mathsf{\{exp\}.\,exp})\mathsf{exp}$$
$$\Upsilon, u : \mathsf{exp} \Vdash \mathsf{get}[u] : ()\mathsf{exp}$$
$$\Upsilon, u : \mathsf{exp} \Vdash \mathsf{set}[u] : (\mathsf{.\,exp})\mathsf{exp}$$

Declaring a new assignable consists in providing an initial value, and an expression binding a symbol (which shall represent the assignable in scope). Note that the functoriality of $\mathscr{O}$ guarantees for any renaming $\varrho : \Upsilon \to \Upsilon'$ and $f : a \to a'$, a lifted map $\mathscr{O}\langle \varrho, f\rangle : \mathscr{O}\langle \Upsilon, a\rangle \to \mathscr{O}\langle \Upsilon', a'\rangle$ such that $\Upsilon' \Vdash \mathscr{O}\langle \varrho, f\rangle(\vartheta) : a'$ when $\Upsilon \Vdash \vartheta : a$. Because the category $\mathscr{A}_{\equiv}$ is discrete and has only identity arrows, through an abuse of notation we will often write $\mathscr{O}\langle \varrho\rangle$ to mean $\mathscr{O}\langle \varrho, 1\rangle$. In particular, the renaming $\Upsilon, u \mapsto \Upsilon, v$ shall take $\mathsf{get}[u]$ to $\mathsf{get}[v]$.

2 Contexts

In general, we will have three kinds of context: metavariable contexts, variable contexts, and symbol (parameter) contexts. A metavariable context Ω consists of bindings of valences

to metavariables; a variable context Γ is a collection of bindings of sorts to variables, and a parameter context Υ is a collection of bindings of sorts to symbols.

$$\frac{}{\cdot\ mctx} \qquad \frac{\Omega\ mctx \quad v\ valence \quad \mathrm{M} \notin |\Omega|}{\Omega, \mathrm{M} : v\ mctx}$$

$$\frac{}{\cdot\ vctx} \qquad \frac{\Gamma\ vctx \quad s\ sort \quad x \notin |\Gamma|}{\Gamma, x : s\ vctx}$$

$$\frac{}{\cdot\ sctx} \qquad \frac{\Upsilon\ vctx \quad s\ sort \quad u \notin |\Upsilon|}{\Upsilon, u : s\ sctx}$$

3 Abstract Binding Trees

Let the judgment $\Omega \rhd \Upsilon \parallel \Gamma \vdash M : s$ presuppose $\Omega\ mctx$, $\Upsilon\ sctx$, $\Gamma\ vctx$ and $s\ sort$, meaning that M is an abstract binding tree of sort s, with metavariables in Ω, parameters in Υ, and variables in Γ. Let the judgment $\Omega \rhd \Upsilon \parallel \Gamma \vdash E : v$ presuppose $v\ valence$. Then, the syntax of abstract binding trees (abts) is inductively defined in four rules:

$$\frac{\Gamma \ni x : s}{\Omega \rhd \Upsilon \parallel \Gamma \vdash x : s}\ var$$

$$\frac{\begin{array}{l}\Omega \ni \mathrm{M} : \{p_0, \ldots, p_m\}[q_0, \ldots, q_n].s \\ \Upsilon \ni u_i : p_i \quad (i \le m) \\ \Omega \rhd \Upsilon \parallel \Gamma \vdash M_i : q_i \quad (i \le n)\end{array}}{\Omega \rhd \Upsilon \parallel \Gamma \vdash \mathrm{M}\{u_0, \ldots, u_m\}(M_0, \ldots, M_n) : s}\ mvar$$

$$\frac{\begin{array}{l}\Upsilon \Vdash \vartheta : v_1, \ldots, v_n \\ \Omega \rhd \Upsilon \parallel \Gamma \vdash E_i : q_i \quad (i \le n)\end{array}}{\Omega \rhd \Upsilon \parallel \Gamma \vdash \vartheta(E_0, \ldots, E_n) : s}\ app$$

$$\frac{\Omega \rhd \Upsilon, \vec{u} : \vec{p} \parallel \Gamma, \vec{x} : \vec{q} \vdash M : s}{\Omega \rhd \Upsilon \parallel \Gamma \vdash \left(\{\vec{u}\}[\vec{x}].M\right) : \left(\{\vec{p}\}[\vec{q}].s\right)}\ abs$$

Abts are identified up to α-equivalence. Let $\mathbf{FS}(M)$ be the collection of symbols free in M, and let $\mathbf{FV}(M)$ be the collection of variables free $\pi_{\mathscr{O}}$ to operators in M.

3.1 Renaming of Symbols

The only place that symbols appear in our calculus is as parameters to operators (unlike variables, symbols are not terms). Therefore, the functorial action of the operator presheaf can be lifted into terms by recursion on their structure, via a pair of judgments $M \restriction \varrho \rightsquigarrow N$ and $E \restriction \varrho \rightsquigarrow F$, presupposing $\varrho : \Upsilon \to \Upsilon'$, and $\Omega \rhd \Upsilon \parallel \Gamma \vdash M : s$ and $\Omega \rhd \Upsilon \parallel \Gamma \vdash E : v$ respectively:

$$\frac{}{x \restriction \varrho \rightsquigarrow x}$$

$$\frac{\varrho(\vec{u}) \equiv \vec{v} \quad M_i \restriction \varrho \rightsquigarrow N_i \quad (i \leq n)}{\mathrm{M}\{\vec{u}\}(M_0, \ldots, M_n) \restriction \varrho \rightsquigarrow \mathrm{M}\{\vec{v}\}(N_0, \ldots, N_n)}$$

$$\frac{\mathscr{O}\langle\varrho\rangle(\vartheta) \equiv \vartheta' \quad E_i \restriction \varrho \rightsquigarrow F_i \quad (i \leq n)}{\vartheta(E_0, \ldots, E_n) \restriction \varrho \rightsquigarrow \vartheta'(F_0, \ldots, F_n)}$$

$$\frac{M \restriction \varrho \setminus \vec{u} \rightsquigarrow N}{\left(\{\vec{u}\}[\vec{x}].\, M\right) \restriction \varrho \rightsquigarrow \left(\{\vec{u}\}[\vec{x}].\, N\right)}$$

Above, the notation $\varrho \setminus \vec{u}$ means the omission of the variables $\vec{u}$ from the renaming ϱ. Because terms are identified up to α-equivalence, the renaming judgment is functional in its input, and so we are justified in writing $M \restriction \varrho$ for N when $M \restriction \varrho \rightsquigarrow N$.

3.2 Substitution of Variables

Variable substitution in abts is defined inductively by a pair of judgments, $[N/x]M \rightsquigarrow M'$ and $[N/x]E \rightsquigarrow F$:

$$\frac{x = y}{[N/x]\, y \rightsquigarrow N} \qquad \frac{x \# y}{[N/x]\, y \rightsquigarrow y}$$

$$\frac{[N/x]\, M_i \rightsquigarrow M_i' \quad (i \leq n)}{[N/x]\, \mathrm{M}\{\vec{u}\}(M_0, \ldots, M_n) \rightsquigarrow \mathrm{M}\{\vec{u}\}(M_0', \ldots, M_n')}$$

$$\frac{[N/x]\, E_i \rightsquigarrow F_i \quad (i \leq n)}{[N/x]\, \vartheta(E_0, \ldots, E_n) \rightsquigarrow \vartheta(F_0, \ldots, F_n)}$$

$$\frac{x \notin \vec{y} \quad \vec{u} \# \mathbf{FS}(N) \quad \vec{y} \# \mathbf{FV}(N) \quad [N/x]M \rightsquigarrow M'}{[N/x]\, \{\vec{u}\}[\vec{y}].\, M \rightsquigarrow \{\vec{u}\}[\vec{y}].\, M'} \qquad \frac{x \in \vec{y} \quad \vec{u} \# \mathbf{FS}(N) \quad \vec{y} \# \mathbf{FV}(N)}{[N/x]\, \{\vec{u}\}[\vec{y}].\, M \rightsquigarrow \{\vec{u}\}[\vec{y}].\, M}$$

Going forward, we will write $[N/x]M$ for M' when $[N/x]M \rightsquigarrow M'$, and $\left[\vec{N}/\vec{x}\right]M$ for the simultaneous substitution of $\vec{N}$ for $\vec{x}$ in M.

3.3 Substitution of Metavariables

Metavariables are substituted by bound terms; since a metavariable may only appear in an application expression $\mathrm{M}\{\cdots\}(\cdots)$, we will instantiate the bound term at the supplied parameters and arguments. Substitution for metavariables is defined inductively by the judgments $[E/\mathrm{M}]M \rightsquigarrow N$ and $[E/\mathrm{M}]F \rightsquigarrow F'$:

$$\frac{}{[E/\mathrm{M}]\,x \rightsquigarrow x}$$

$$\frac{\mathrm{M} \# \mathrm{N} \quad [E/\mathrm{N}]M_i \rightsquigarrow N_i \quad (i \leq n)}{[E/\mathrm{M}]\,\mathrm{N}\{\vec{u}\}(M_0, \ldots, M_n) \rightsquigarrow \mathrm{N}\{\vec{u}\}(N_0, \ldots, N_n)}$$

$$\frac{\mathrm{M} = \mathrm{N} \quad \left[\vec{M}/\vec{x}\right]N \rightsquigarrow N' \quad N' \restriction \left\{\vec{u} \mapsto \vec{v}\right\} \rightsquigarrow N''}{\left[\{\vec{u}\}[\vec{x}].\,N/\mathrm{M}\right]\mathrm{N}\{\vec{v}\}(\vec{M}) \rightsquigarrow N''}$$

$$\frac{[E/\mathrm{M}]F_i \rightsquigarrow F_i' \quad (i \leq n)}{[E/\mathrm{M}]\,\vartheta(F_0, \ldots, F_n) \rightsquigarrow \vartheta(F_0', \ldots, F_n')}$$

$$\frac{\vec{u} \# \mathbf{FS}(E) \quad \vec{x} \# \mathbf{FV}(E) \quad [E/\mathrm{M}]M \rightsquigarrow N}{[E/\mathrm{M}]\,\{\vec{u}\}[\vec{x}].\,M \rightsquigarrow \{\vec{u}\}[\vec{x}].\,N}$$

As usual, we will write $[E/\mathrm{M}]M$ for N when $[E/\mathrm{M}]M \rightsquigarrow N$.

4 Case Study: Wellformed Sequents

The representation of telescopes and sequents in a logical framework is notoriously difficult; whilst it is possible to use higher-order abstract syntax or abts to encode the binding-structure of telescopes and sequents, the encoding is sufficiently laborious and obscure that it is not used in practice.

Crary has demonstrated a first-order encoding of contexts in the logical framework in bijection with actual LF-contexts, which has been successfully used in large-scale mechanization efforts, including that of Standard ML and the Edinburgh Logical Framework itself.

We will approach the problem of encoding telescopes and sequents from the *refinements* perspective, where a conservative approximation of the grammar is first given using the abt logical framework, and then the correctness of a code is expressed separately in a judgment that refines the existing specification.

Because we have not committed to using the built-in binding machinery to express the well-scopedness of telescopes and sequents, we are free to use *symbols* in order to model the variables in the context. This is actually quite sensible if we are trying to actually faithfully represent the syntax of telescopes and sequents, rather than replace them with

their counterparts on the meta-level. It is certainly not intended that we should be able to take a context $\Gamma \equiv x : A$ and apply the substitution $[M/x]\,\Gamma \rightsquigarrow M : A$, which is nonsensical; on the other hand, if x had been a *symbol*, all we could do is rename it in the context, since substitution is not defined for symbols.

This insight leads the way to a simple abt signature for the theory of telescopes and sequents.

$$\begin{array}{l}
\mathsf{tele}\ \mathit{sort} \\
\mathsf{exp}\ \mathit{sort} \\
\mathsf{type}\ \mathit{sort} \\
\mathsf{jdg}\ \mathit{sort}
\end{array}$$

$$\Upsilon, u : \mathsf{exp} \Vdash \mathsf{var}[u] : ()\mathsf{exp}$$

$$\begin{array}{l}
\Upsilon \Vdash \mathsf{nil} : ()\mathsf{tele} \\
\Upsilon, u : \mathsf{exp} \Vdash \mathsf{snoc}[u] : (.\,\mathsf{tele}, .\,\mathsf{type})\mathsf{tele}
\end{array}$$

$$\Upsilon \Vdash \mathsf{sequent} : (.\,\mathsf{tele}, .\,\mathsf{type})\mathsf{jdg}$$

Suppose we have defined encoded a fragment of type theory as well:

$$\begin{array}{l}
\Upsilon \Vdash \top : ()\mathsf{type} \\
\Upsilon \Vdash \bot : ()\mathsf{type} \\
\Upsilon \Vdash \mathsf{bool} : ()\mathsf{type} \\
\Upsilon \Vdash \mathsf{isTrue} : (.\,\mathsf{exp})\mathsf{type} \\
\Upsilon \Vdash \mathsf{pi} : (.\,\mathsf{type}, [\mathsf{exp}].\,\mathsf{type})\mathsf{type} \\
\Upsilon \Vdash \mathsf{sg} : (.\,\mathsf{type}, [\mathsf{exp}].\,\mathsf{type})\mathsf{type}
\end{array}$$

Terms written using the abstract syntax will be difficult to read, so let us define some notation:

$$\begin{array}{rcl}
\lozenge & \triangleq & \mathsf{nil} \\
\Gamma, u : P & \triangleq & \mathsf{snoc}[u](\Gamma, P) \\
\Gamma \gg P & \triangleq & \mathsf{sequent}(\Gamma, P) \\
\grave{}u & \triangleq & \mathsf{var}[u]
\end{array}$$

Now, we have the following well-formed sequent:

$$\cdot \triangleright u : \mathsf{exp}, v : \mathsf{exp} \parallel \cdot \vdash \lozenge, u : \mathsf{bool}, v : \mathsf{isTrue}(\grave{}u) \gg \mathsf{isTrue}(\grave{}u) : \mathsf{jdg}$$

The above sequent has free symbols, but we can close over them by adding a form of parametric higher-order judgment to our object language, indexed by a collection of sorts $\vec{s}$:

$$\Upsilon \Vdash \nabla[\vec{s}] : (\{\vec{s}\}.\,\mathsf{jdg})\mathsf{jdg}$$

Then, we may write a closed sequent judgment as follows:

$$\cdot \rhd \cdot \parallel \cdot \vdash \nabla[\mathsf{exp}, \mathsf{exp}](\{u, v\}[]. \diamond, u : \mathsf{bool}, v : \mathsf{isTrue}(\text{`}u) \gg \mathsf{isTrue}(\text{`}u)) : \mathsf{jdg}$$

4.1 Refinements for wellformedness

Having specified an approximation of the grammar of telescopes and sequents in the abt logical framework, we can proceed to define proper wellformedness via *inductive refinement*. The basic idea is to introduce a new form of (meta)-judgment $\Omega \rhd \Upsilon \parallel \Gamma \vdash M \in_{\mathbf{wf}} s$ which expresses the extrinsic wellformedness properties we wish to verify, presupposing $\Omega \rhd \Upsilon \parallel \Gamma \vdash M : s$.