

Abstract Binding Trees

Jon Sterling and Darin Morrison

1 Preliminaries

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

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

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

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

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 $\mathbf{Sctx} \triangleq \mathbb{F} \downarrow \mathcal{S}$, with \mathcal{S} regarded as a discrete category, is the category of contexts of symbols, whose objects are finite sets of symbols U and sort-assignments $\mathfrak{s} : U \rightarrow \mathcal{S}$, and whose morphisms are sort-preserving renamings; we will write Υ for a symbol context (U, \mathfrak{s}) .

Then, fix a covariant presheaf of operators $\mathcal{O} : \mathbf{Sctx} \times \mathcal{A} \rightarrow \mathbf{Set}$ such that the arrows in \mathbf{Sctx} lift to renamings of operators' parameters. Via the Grothendieck construction¹ \oint on the operator presheaf \mathcal{O} we also have a category with objects $\langle \Upsilon, \varrho \rangle, \vartheta \in \oint \mathcal{O}$ for $\vartheta \in \mathcal{O}(\Upsilon, \varrho)$ and morphisms $\oint \mathcal{O} [\langle \Upsilon, \varrho \rangle, \vartheta, \langle \Upsilon', \varrho' \rangle, \vartheta']$ for $f : \langle \Upsilon, \varrho \rangle \rightarrow \langle \Upsilon', \varrho' \rangle$ such that $f^* \vartheta = \vartheta' \in \mathcal{O}(\Upsilon', \varrho')$.

$$\frac{\vartheta \in \mathcal{O}(\Upsilon, \varrho)}{\Upsilon \Vdash \vartheta : \varrho}$$

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

¹Category of elements.

Examples Operators are defined by specifying the fibres of $\oint \mathcal{C}$ 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:

$$\begin{aligned} \Upsilon \Vdash \lambda &: (\{\cdot\} [\text{exp}] . \text{exp}) \text{exp} \\ \Upsilon \Vdash \text{ap} &: (\{\cdot\} [\cdot] . \text{exp}, \{\cdot\} [\cdot] . \text{exp}) \text{exp} \end{aligned}$$

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

$$\begin{aligned} \Upsilon \Vdash \text{decl} &: (\{\cdot\} [\cdot] . \text{exp}, \{\text{exp}\} [\cdot] . \text{exp}) \text{exp} \\ \Upsilon, u : \text{exp} \Vdash \text{get}[u] &: (\cdot) \text{exp} \\ \Upsilon, u : \text{exp} \Vdash \text{set}[u] &: (\{\cdot\} [\cdot] . \text{exp}) \text{exp} \end{aligned}$$

Declaring a new assignable consists in providing an initial value, and an expression with a free symbol (which shall represent the assignable in scope). Weakening can be seen as inducing a “degeneracy map” on operators, whereas a renaming $u \mapsto v$ will take $\text{get}[u]$ to $\text{get}[v]$.

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.

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

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

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

3 Abstract Binding Trees

Let the judgment $\Omega \triangleright \Upsilon \parallel \Gamma \vdash M : s$ presuppose Ω *mctx*, Υ *sctx*, Γ *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 $\Omega \triangleright \Upsilon \parallel \Gamma \vdash E : v$ presuppose v *valence*. Then, the syntax of abstract binding trees is inductively defined in four rules:

$$\begin{array}{c}
\frac{\Gamma \ni x : s}{\Omega \triangleright \Upsilon \parallel \Gamma \vdash x : s} \textit{var} \\[10pt]
\frac{\begin{array}{l} \Omega \ni M : \{p_0, \dots, p_m\} [q_0, \dots, q_n]. s \\ \Upsilon \ni u_i : p_i \quad (i \leq m) \\ \Omega \triangleright \Upsilon \parallel \Gamma \vdash M_i : q_i \quad (i \leq n) \end{array}}{\Omega \triangleright \Upsilon \parallel \Gamma \vdash M \{u_0, \dots, u_m\} (M_0, \dots, M_n) : s} \textit{mvar} \\[10pt]
\frac{\begin{array}{l} \Upsilon \Vdash \vartheta : v_1, \dots, v_n \\ \Omega \triangleright \Upsilon \parallel \Gamma \vdash E_i : q_i \quad (i \leq n) \end{array}}{\Omega \triangleright \Upsilon \parallel \Gamma \vdash \vartheta (E_0, \dots, E_n) : s} \textit{app} \\[10pt]
\frac{\Omega \triangleright \Upsilon, \vec{u} : \vec{p} \parallel \Gamma, \vec{x} : \vec{q} \vdash M : s}{\Omega \triangleright \Upsilon \parallel \Gamma \vdash (\{\vec{u}\} [\vec{x}]. M) : (\{\vec{p}\} [\vec{q}]. s)} \textit{abs}
\end{array}$$

Abstract binding trees are identified up to α -equivalence.

3.1 Substitution of Variables

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

$$\begin{array}{c}
\frac{x = y}{[N/x]y \rightsquigarrow N} \quad \frac{x \# y}{[N/x]y \rightsquigarrow y} \\[10pt]
\frac{[N/x]M_i \rightsquigarrow M'_i \quad (i \leq n)}{[N/x]M \{\vec{u}\} (M_0, \dots, M_n) \rightsquigarrow M \{\vec{u}\} (M'_0, \dots, M'_n)} \\[10pt]
\frac{[N/x]E_i \rightsquigarrow E'_i \quad (i \leq n)}{[N/x]\vartheta (E_0, \dots, E_n) \rightsquigarrow \vartheta (E'_0, \dots, E'_n)} \\[10pt]
\frac{x \notin \vec{y} \quad \vec{y} \# \mathbf{FV}(N) \quad [N/x]M \rightsquigarrow M'}{[N/x]\{\vec{u}\} [\vec{y}]. M \rightsquigarrow \{\vec{u}\} [\vec{y}]. M'} \quad \frac{x \in \vec{y} \quad \vec{y} \# \mathbf{FV}(N)}{[N/x]\{\vec{u}\} [\vec{y}]. M \rightsquigarrow \{\vec{u}\} [\vec{y}]. M}
\end{array}$$

Because terms are identified up to α -equivalence, the variable substitution judgment is functional in its inputs, and so we are justified in writing $[N/x]M$ for M' when $[N/x]M \rightsquigarrow M'$. We write $[\vec{N}/\vec{x}]M$ for the simultaneous substitution of \vec{N} for \vec{x} in M .

3.2 Substitution of Metavariables

Metavariable substitution is defined inductively by the judgment $[E/M]M \rightsquigarrow M'$:

$$\frac{\overline{[E/M]x \rightsquigarrow x} \quad \begin{array}{c} M \# N \quad [E/N]M_i \rightsquigarrow M'_i \quad (i \leq n) \end{array}}{[E/M]N\{\vec{u}\}(M_0, \dots, M_n) \rightsquigarrow N\{\vec{u}\}(M'_0, \dots, M'_n)}$$