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 $\vec{q}$.

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

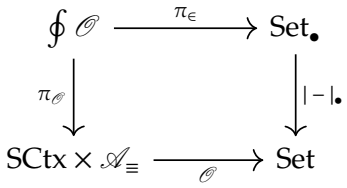
An arity $(\vec{v})\, s$ specifies an operator of sort s with arguments of valences $\vec{v}$. We will call the set of valences $\mathscr{V}$, and the set of arities $\mathscr{A}$.

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

Let I be an infinite set of symbols. Let $\mathbb{F}_{\mathrm{I}}$ be the category of finite subsets of I and their injective maps; then the comma construction $\mathsf{SCtx} \triangleq \mathbb{F}_{\mathrm{I}} \downarrow \mathscr{S}_{\equiv}$, 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 $\mathscr{O} : \mathsf{SCtx} \times \mathscr{A}_{\equiv} \to \mathsf{Set}$ such that the arrows in SCtx lift to renamings of operators' parameters. Via the Grothendieck construction[1] $\oint (-) : \mathsf{Set}^{\mathsf{C}} \to \mathsf{Cat}$ 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 $\left\langle \varrho, f \right\rangle : \langle \Upsilon, a \rangle \to \langle \Upsilon', a' \rangle$ 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 Set-bundle where $\pi_{\mathscr{O}}$ is a discrete Grothendieck opfibration and $|-|_\bullet$ is the forgetful functor from pointed sets:

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



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

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:

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

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

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

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 $\mathsf{get}[v]$.

Definition 1.1. A *signature* Σ is a set of sorts $\mathscr{S}$ together with a copresheaf $\mathscr{O}$ of operators.

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.

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

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

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

3 Abstract Binding Trees

Let the judgment $\Omega \rhd \Upsilon \parallel \Gamma \vdash M : s$ presuppose $\Omega\ \mathit{mctx}$, $\Upsilon\ \mathit{sctx}$, $\Gamma\ \mathit{vctx}$ and $s\ \mathit{sort}$, 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 $v\ \mathit{valence}$. Then, the syntax of abstract binding trees (abts) is inductively defined in four rules:

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

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

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

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

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

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

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

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

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

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 $[N/x]\, E \rightsquigarrow F$:

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

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

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

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

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 $[E/\mathrm{m}]\,F \rightsquigarrow F'$:

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

$$\frac{\mathrm{m} \# \mathrm{n} \quad [E/\mathrm{n}]\,M_i \rightsquigarrow N_i \;\; (i \le n)}{[E/\mathrm{m}]\,\mathrm{n}\{\vec{u}\}(M_0, \dots, M_n) \rightsquigarrow \mathrm{n}\{\vec{u}\}(N_0, \dots, N_n)}$$

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

$$\frac{[E/\mathrm{m}]\,F_i \rightsquigarrow F_i' \;\; (i \le n)}{[E/\mathrm{m}]\,\vartheta(F_0, \dots, F_n) \rightsquigarrow \vartheta(F_0', \dots, F_n')}$$

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

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

4 Model Theory

Let $\mathbb{F}$ be the free cocartesian category on a single object (that is, the category of finite cardinals and functions between them); then the comma construction $\mathrm{Ctx} \triangleq \mathbb{F} \downarrow \mathscr{S}$ is the category of variable contexts. Let $\mathrm{H} \triangleq (\mathrm{SCtx} \times \mathrm{Ctx})^{\mathrm{op}}$; then we fix the functor category $\widehat{\mathrm{H}}^{\mathscr{S}}$ as our semantic universe.

Let $V_s(\Upsilon \parallel \Gamma) \triangleq \{x \in |\Gamma| \mid \Gamma(x) = s\}$ be called the presheaf of variables; additionally, we have a presheaf of symbols $S_s(\Upsilon \parallel \Gamma) \triangleq \{u \in |\Upsilon| \mid \Upsilon(x) = s\}$. Lastly, we have the presheaf of operators with arity a, $\mathscr{O}_a(\Upsilon \parallel \Gamma) \triangleq \mathscr{O}\,\langle \Upsilon, a \rangle$

4.1 Substitution monoidal structures

For an object $P : \widehat{\mathrm{H}}^{\mathscr{S}}$, we will use the notation $P^{[\Gamma]}$ to mean $\prod_{x \in |\Gamma|} P_{\Gamma(x)}$; likewise, $S^{\{\Upsilon\}}$ shall mean $\prod_{u \in |\Upsilon|} S_{\Upsilon(u)}$.

For a presheaf $A : \widehat{\mathrm{H}}$ and a sort-indexed family of presheaves $P : \widehat{\mathrm{H}}^{\mathscr{S}}$, we have an operation $A \bullet P$, defined as a coend in the following way:

$$(A \bullet P)(\Upsilon \parallel \Gamma) \triangleq \int^{(\Upsilon' \parallel \Delta) \in \mathrm{H}} A(\Upsilon' \parallel \Delta) \times S^{\{\Upsilon'\}}(\Upsilon \parallel \Gamma) \times P^{[\Delta]}(\Upsilon \parallel \Gamma)$$

Using this, we can define a tensor $P \odot Q$ for $P, Q : \widehat{\mathrm{H}}^{\mathscr{S}}$ as follows:

$$(P \odot Q)_s \triangleq P_s \bullet Q \qquad (s \in \mathscr{S})$$

Then, V is the unit to this tensor. We will say that an object $P : \widehat{\mathrm{H}}^{\mathscr{S}}$ is a Σ-monoid in case it is equipped with the following natural transformations where ν embeds variables into P and ς equips P with an operation for simultaneous substitutions of variables.

$$V \xrightarrow{\ \nu\ } P \xleftarrow{\ \varsigma\ } P \odot P$$

We also have an induced maps $\nu_\Gamma : V^{[\Gamma]} \to P^{[\Gamma]}$ and $\varsigma_{\Upsilon \| \Gamma} : P_s^{y(\Upsilon \| \Gamma)} \times S^{\{\Upsilon\}} \times P^{[\Gamma]} \to P_s$.

Remark 4.1. A presheaf P with a substitution monoidal structure $\langle \nu, \varsigma \rangle$ also forms a *relative monad*[2] on the functor $V : \widehat{\mathrm{H}}^{\mathscr{S}}$ of variables. Consequently, the monoid multiplication ς can be replaced with an extension operation:

$$P(=)^{V(-)} \xrightarrow{\ (-)^{\circledast}\ } P(=)^{P(-)}$$

4.2 The signature endofunctor and its initial algebras

For each signature $\Sigma \equiv \langle \mathscr{S}, \mathscr{O} \rangle$, we have an endofunctor $\mathscr{F}_\Sigma : \widehat{\mathrm{H}}^{\mathscr{S}} \to \widehat{\mathrm{H}}^{\mathscr{S}}$, which is defined as follows:

$$\mathscr{F}_\Sigma(X)_s \triangleq \coprod_{\vartheta \in \mathscr{O}_{(\vec{v})\, s}} \prod_{[\vec{p}][\vec{q}].\, s' \in \vec{v}} X_{s'}^{y(\vec{p} \| \vec{q})}$$

Then, a Σ-model is a Σ-monoid P which is equipped with an initial algebra $\alpha : \mathscr{F}_\Sigma(P) \to P$, which shall interpret applications of each operator.

4.3 Interpretation of terms

The metavariable, symbol and variable contexts are interpreted in a model P as an environment presheaf in the following way:

$$[\![\Omega \triangleright \Upsilon \parallel \Gamma]\!]_P \triangleq \left(\prod_{(m:[\vec{p}][\vec{q}].\, s) \in \Omega} P_s^{y(\vec{p} \| \vec{q})} \right) \times S^{\{\Upsilon\}} \times V^{[\Gamma]}$$

[2] Relative monads are skew-monoids in the skew-monoidal structure induced by a functor category C^{J} [2]. However, relative monads can also be understood without reference to monoids by viewing them as a generalization of *extension systems* or *Kleisli structures*. This latter presentation of monads, popular in computer science and logic, does not involve iteration of the monad functor.

Then, the interpretation of a term in a model P is a map from its environment to P, that is, $[\![\Omega \rhd \Upsilon \parallel \Gamma \vdash M : s]\!]_P : [\![\Omega \rhd \Upsilon \parallel \Gamma]\!]_P \to P_s$. The interpretation of a variable $\left[\!\!\left[\Omega \rhd \Upsilon \parallel \Gamma \vdash x_j : s\right]\!\!\right]_P$ simply projects it of the environment and embeds it into the model, as follows:

$$[\![\Omega \rhd \Upsilon \parallel \Gamma]\!]_P \xrightarrow{\pi_3} V^{[\Gamma]} \xrightarrow{\nu_\Gamma} P^{[\Gamma]} \xrightarrow{\pi_j} P_s \qquad \text{(var)}$$

The interpretation of metavariable applications $[\![\Omega \rhd \Upsilon \parallel \Gamma \vdash \mathrm{m}_i\{u_0, \ldots, u_m\}(M_0, \ldots, M_n) : s]\!]_P$, where $\Omega \ni \mathrm{m} : \{\vec{p}\}[\vec{q}].\, s$, proceeds by projecting the metavariable's interpretation from the environment and intantiating it via substitution:

$$[\![\Omega \rhd \Upsilon \parallel \Gamma]\!]_P \xrightarrow{\langle \pi_i\, \pi_1, \phi\, \pi_2, \psi \rangle} P_{s'}^{y(\vec{p} \| \vec{q})} \times S^{\{\vec{p}\}} \times P^{[\vec{q}]} \xrightarrow{\varsigma_{\vec{p} \| \vec{q}}} P_s \qquad \text{(metavar)}$$

where

$$\phi \triangleq \left|\vec{p}\right| \mapsto \vec{u}$$
$$\psi \triangleq \left\langle \left[\!\!\left[\Omega \rhd \Upsilon \parallel \Gamma \vdash M_j : q_j\right]\!\!\right]_P \right\rangle_{j \le n}$$

5 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 [3], which has been successfully used in large-scale mechanization efforts, including that of Standard ML [10] and the Edinburgh Logical Framework itself [11].

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 in fact quite sensible if we are actually trying to faithfully represent the syntax of telescopes and sequents, rather than replace them with their counterparts on the meta-level.

This insight leads the way to a simple abt signature for the theory of telescopes and se-

quents.

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

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

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

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

Suppose we have 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:

$$\begin{array}{rl}\diamond & \triangleq \mathsf{nil}\\ H, u : P & \triangleq \mathsf{snoc}[u](H, P)\\ H \gg A & \triangleq \mathsf{sequent}(H, A)\\ `u & \triangleq \mathsf{var}[u]\end{array}$$

Now, we have the following well-formed sequent:

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

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:

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

5.1 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 $\Omega \rhd \Upsilon \parallel \Gamma \vdash M \in_{\mathrm{wf}} s$ which expresses the extrinsic wellformedness properties we wish to verify, presupposing $\Omega \rhd \Upsilon \parallel \Gamma \vdash M : s$. Additionally, we introduce an analogous judgment on bound terms, $\Omega \rhd \Upsilon \parallel \Gamma \vdash E \in_{\mathrm{wf}} v$ presupposing $\Omega \rhd \Upsilon \parallel \Gamma \vdash E : v$, defined uniformly as follows:

$$\frac{\Omega \rhd \Upsilon, \vec{u} : \vec{p} \parallel \Gamma, \vec{x} : \vec{q} \vdash M \in_{\mathrm{wf}} s}{\Omega \rhd \Upsilon \parallel \Gamma \vdash \{\vec{u}\}[\vec{x}].\, M \in_{\mathrm{wf}} \{\vec{p}\}[\vec{q}].\, s}$$

Likewise, wellformedness for variables and metavariables is defined uniformly:

$$\frac{}{\Omega \rhd \Upsilon \parallel \Gamma \vdash x \in_{\mathrm{wf}} s} \qquad \frac{\begin{array}{c}\Omega \ni \mathrm{m} : \{\vec{p}\}[q_0, \ldots, q_n].\, s \\ \Omega \rhd \Upsilon \parallel \Gamma \vdash M_i \in_{\mathrm{wf}} q_i \;\; (i \leq n)\end{array}}{\Omega \rhd \Upsilon \parallel \Gamma \vdash \mathrm{m}\{\vec{u}\}(M_0, \ldots, M_n) \in_{\mathrm{wf}} s}$$

Remark 5.1. Note that the refinement for variables x is not trivial, since it is only defined in case the presupposition $\Omega \rhd \Upsilon \parallel \Gamma \vdash x : s$ is satisfied.

The remainder of the definition of refinement proceeds by induction on sorts and operators. For the sake of this example, we will just stipulate that anything of sort exp or type is grammatical if its subterms are grammatical:

$$\frac{\begin{array}{c}\Upsilon \Vdash \vartheta : (v_0, \ldots, v_n)\, s \\ \Omega \rhd \Upsilon \parallel \Gamma \vdash E_i \in_{\mathrm{wf}} v_i \;\; (i \leq n)\end{array}}{\Omega \rhd \Upsilon \parallel \Gamma \vdash \vartheta(E_0, \ldots, E_n) \in_{\mathrm{wf}} s} \text{ for } s \in \{\mathrm{exp}, \mathrm{type}\}$$

The refinements for parametric judgment and sequents simply delegate to their subterms as well:

$$\frac{\Omega \rhd \Upsilon, \vec{u} : \vec{s} \parallel \Gamma \vdash J \in_{\mathrm{wf}} \mathrm{jdg}}{\Omega \rhd \Upsilon \parallel \Gamma \vdash \nabla[\vec{s}](\{\vec{u}\}[].\, J) \in_{\mathrm{wf}} \mathrm{jdg}}$$

$$\frac{\Omega \rhd \Upsilon \parallel \Gamma \vdash H \in_{\mathrm{wf}} \mathrm{tele} \quad \Omega \rhd \Upsilon \parallel \Gamma \vdash A \in_{\mathrm{wf}} \mathrm{type}}{\Omega \rhd \Upsilon \parallel \Gamma \vdash H \gg A \in_{\mathrm{wf}} \mathrm{jdg}}$$

The refinement for telescopes proceeds by induction:

$$\frac{}{\Omega \rhd \Upsilon \parallel \Gamma \vdash \diamond \in_{\mathrm{wf}} \mathrm{tele}} \qquad \frac{\begin{array}{c}\Omega \rhd \Upsilon \setminus \{u\} \parallel \Gamma \vdash H \in_{\mathrm{wf}} \mathrm{tele} \\ \Omega \rhd \Upsilon \setminus \{u\} \parallel \Gamma \vdash A \in_{\mathrm{wf}} \mathrm{type}\end{array}}{\Omega \rhd \Upsilon \parallel \Gamma \vdash H, u : A \in_{\mathrm{wf}} \mathrm{tele}}$$

References

[1] P. Aczel. A general Church-Rosser theorem. Technical report, University of Manchester, 1978.

[2] T. Altenkirch, J. Chapman, and T. Uustalu. Monads need not be endofunctors. *Logical Methods in Computer Science*, 11(1:3):1–40, 2015.

[3] K. Crary. Explicit contexts in LF (extended abstract). *Electronic Notes in Theoretical Computer Science*, 228:53 – 68, 2009. Proceedings of the International Workshop on Logical Frameworks and Metalanguages: Theory and Practice (LFMTP 2008).

[4] M. Fiore and O. Mamoud. Second-order algebraic theories – (extended abstract). In *Mathematical Foundations of Computer Science 2010, 35th International Symposium, MFCS 2010, Brno, Czech Republic, August 23-27, 2010. Proceedings*, pages 368–380, 2010.

[5] M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Proceedings of the 14th Symposium on Logic in Computer Science*, pages 193–202, 1999.

[6] M. P. Fiore. Mathematical models of computational and combinatorial structures. In *Foundations of Software Science and Computational Structures, 8th International Conference, FOSSACS 2005, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005, Edinburgh, UK, April 4-8, 2005, Proceedings*, pages 25–46, 2005.

[7] M. Hamana. Free Σ-monoids: A higher-order syntax with metavariables. In W.-N. Chin, editor, *Programming Languages and Systems*, volume 3302 of *Lecture Notes in Computer Science*, pages 348–363. Springer Berlin Heidelberg, 2004.

[8] R. Harper. *Practical Foundations for Programming Languages*. Cambridge University Press, New York, NY, USA, 2016.

[9] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *J. ACM*, 40(1):143–184, Jan. 1993.

[10] D. K. Lee, K. Crary, and R. Harper. Towards a mechanized metatheory of standard ml. In *Proceedings of the 34th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '07, pages 173–184, New York, NY, USA, 2007. ACM.

[11] C. Martens and K. Crary. LF in LF: Mechanizing the metatheories of LF in Twelf. In *Proceedings of the Seventh International Workshop on Logical Frameworks and Meta-languages, Theory and Practice*, LFMTP '12, pages 23–32, New York, NY, USA, 2012. ACM.

[12] P. Martin-Löf and G. Sambin. *Intuitionistic type theory*. Studies in proof theory. Bibliopolis, Napoli, 1984.