1 Syntax

$$
\begin{array}{rcl}
e,\ p & ::= & \\
& | & x \\
& | & <> \\
& | & \lambda^{\Pi} x : A.\, e \\
& | & \lambda^{\forall} \underline{x} : A.\, e \\
& | & e_1\, e_2 \\
& | & e_1\{e_2\} \\
& | & \mathbf{refl}\, e \\
& | & \mathbf{subst}\, (p : e_1 = e_2 : A, x.\, B, e) \\
& | & \mathbf{let}\, x : A = e_1\, \mathbf{in}\, e_2 \\
& | & \widehat{x}[\sigma] \\
& | & \_ \\
A,\ B,\ C,\ D & ::= & \\
& | & 1 \\
& | & \Pi x : A.\, B \\
& | & \forall \underline{x} : A.\, B \\
& | & e_1 = e_2 : A \\
& | & \mathbf{let}\, x : A = e\, \mathbf{in}\, B
\end{array}
$$

2 Safety

To formulate the completeness properties (at some point in the future), we need to restrict the 'implicit' binders $\lambda^\forall$ and $\forall$ to bind variables that are at safe positions in the body. For this purpose, we define the well-formedness relation.

Here V is a set of variables, including normal variables (x) and safe variables ($\underline{x}$), whose positions are restricted. Overall, we wish to guarantee that after normalization, substitution of safe variables does not produce new redexes.

The constructor forms are well-formed by congruence: if their components are well-formed. Notice that $\lambda^\forall$ and $\forall$ require the bound variable to be safe in the body.

The eliminator forms are well-formed if their components are well-formed and one of the following conditions hold:

- they do not contain variables that are required to be safe
- they are *inert* (see below), which implies that their outer structure withstands reduction and substitution of safe variables.
- they are reducible but the reduction preserves safety: it does not eliminate metavariables, and if the substitution happens, the variables we are substituting are at safe positions and the terms we are substituting for are safe.

$\boxed{V \vdash e\ \mathbf{OK}}$

$$\frac{x \in V}{V \vdash x\ \mathbf{OK}}\ \textsc{OkOKVar}$$

$$\frac{}{V \vdash \widehat{x}[\sigma]\ \mathbf{OK}}\ \textsc{OkOKMVar}$$

$$\frac{}{V \vdash <>\ \mathbf{OK}}\ \textsc{OkOKUnit}$$

$$\frac{V \vdash A\ \mathbf{OK} \quad V, x \vdash e\ \mathbf{OK}}{V \vdash \lambda^\Pi x : A.\, e\ \mathbf{OK}}\ \textsc{OkOKLam}$$

$$\frac{V \vdash A\ \mathbf{OK} \quad V, \underline{x} \vdash e\ \mathbf{OK} \quad \underline{x} \in \mathbf{fv}\, e}{V \vdash \lambda^\forall \underline{x} : A.\, e\ \mathbf{OK}}\ \textsc{OkOKPLam}$$

$$\frac{V \vdash e\ \mathbf{OK}}{V \vdash \mathbf{refl}\, e\ \mathbf{OK}}\ \textsc{OkOKRefl}$$

$$\frac{e_1, e_2\ \mathbf{are\ ground} \quad V \vdash e_1, e_2\ \mathbf{OK}}{V \vdash e_1\, e_2\ \mathbf{OK}}\ \textsc{OkOKAppGround}$$

$$\frac{V \vdash e_1, e_2\ \mathbf{OK} \quad (e_1\, e_2)\ \mathbf{INERT}}{V \vdash e_1\, e_2\ \mathbf{OK}}\ \textsc{OkOKAppInert}$$

$$\frac{V \vdash \lambda^\forall \underline{x} : A.\, [\underline{x}/x]e_1\ \mathbf{OK} \quad A\ \mathbf{is\ ground} \quad V \vdash e_2\ \mathbf{OK}}{V \vdash (\lambda^\Pi x : A.\, e_1)\, e_2\ \mathbf{OK}}\ \textsc{OkOKAppOK}$$

$$\frac{V \vdash \lambda^\Pi x : A.\, e_1\ \mathbf{OK} \quad A\ \mathbf{is\ ground} \quad V \vdash e_2\ \mathbf{OK} \quad e_2\ \mathbf{NeuINERT}}{V \vdash (\lambda^\Pi x : A.\, e_1)\, e_2\ \mathbf{OK}}\ \textsc{OkOKAppNeut}$$

$$\frac{V \vdash \lambda^{\forall} \underline{x} : A.\, e_1\ \mathbf{OK} \quad A\ \mathbf{is\ ground} \quad V \vdash e_2\ \mathbf{OK}}{V \vdash (\lambda^{\forall} \underline{x} : A.\, e_1)\{e_2\}\ \mathbf{OK}}\ \textsc{OkOKAppSafe}$$

$$\frac{V \vdash e_1, e_2, p, e, A\ \mathbf{OK} \quad V, x \vdash B\ \mathbf{OK} \quad e_1, e_2, p, A, B\ \mathbf{are\ ground}}{V \vdash \mathbf{subst}\ (p : e_1 = e_2 : A, x.\, B, e)\ \mathbf{OK}}\ \textsc{OkOKSubstGround}$$

$$\frac{V \vdash e_1, e_2, p, e, A\ \mathbf{OK} \quad V, x \vdash B\ \mathbf{OK} \quad \mathbf{subst}\ (p : e_1 = e_2 : A, x.\, B, e)\ \mathbf{INERT}}{V \vdash \mathbf{subst}\ (p : e_1 = e_2 : A, x.\, B, e)\ \mathbf{OK}}\ \textsc{OkOKSubstInert}$$

The types are not eliminated, so they behave as constructor forms: they are well-formed by congruence.

The variables are always inert. The constructor forms are always inert.

Inert term preserve their outer structure under reduction and substitution of safe variables. Neutrally inert term, in addition, cannot be reduced when put in a redex position.

$\boxed{e\ \mathbf{NeuINERT}}$

$$\frac{}{x\ \mathbf{NeuINERT}}\ \textsc{OkNeuInertVar}$$

$$\frac{e_1\ \mathbf{NeuINERT}}{e_1\ e_2\ \mathbf{NeuINERT}}\ \textsc{OkNeuInertApp}$$

$$\frac{p\ \mathbf{NeuINERT}}{\mathbf{subst}\ (p : e_1 = e_2 : A, x.\, B, e)\ \mathbf{NeuINERT}}\ \textsc{OkNeuInertSubst}$$

$\boxed{e\ \mathbf{INERT}}$

$$\frac{e\ \mathbf{NeuINERT}}{e\ \mathbf{INERT}}\ \textsc{OkInertNeu}$$

$$\frac{}{<>\ \mathbf{INERT}}\ \textsc{OkInertUnit}$$

$$\frac{}{\lambda^{\Pi} x : A.\, e\ \mathbf{INERT}}\ \textsc{OkInertLam}$$

$$\frac{}{\lambda^{\forall} \underline{x} : A.\, e\ \mathbf{INERT}}\ \textsc{OkInertPLam}$$

$$\frac{}{\mathbf{refl}\ e\ \mathbf{INERT}}\ \textsc{OkInertRefl}$$

3 Reduction

The reduction is defined in a standard way: we first reduce the components of terms (congruence rules), then we eliminate the redexes by the following rules. $\boxed{e_1 \rightsquigarrow e_2}$

$$\frac{}{(\lambda^{\Pi} x : A.\, e_0)\; e \rightsquigarrow [e/x]e_0} \quad \textsc{RedRedAppLam}$$

$$\frac{}{(\lambda^{\forall} \underline{x} : A.\, e_0)\{e\} \rightsquigarrow [e/\underline{x}]e_0} \quad \textsc{RedRedAppPLam}$$

$$\frac{}{\mathbf{subst}\;((\mathbf{refl}\; e_0) : e_1 = e_2 : A, x.\, B, e) \rightsquigarrow e} \quad \textsc{RedRedSubstRed}$$

The invariants that we are preserving:

- reduction does not eliminate metavariables and the safe variables;
- reduction preserves safety of the metavariables and the safe variables.

3.1 Properties

Lemma 1 (Reduction preserves intertness)**.**

Lemma 2 (Safety is preserved when a safe term plugs into a safe position)**.**

Lemma 3 (Safety is preserved when a safe and neutrally inert term plugs into any position)**.**

Lemma 4 (Reduction preserves safety)**.** *If* $V \vdash e_0$ **OK** *and* $e_0 \rightsquigarrow e_1$ *then* $V \vdash e_1$ **OK**. *If* $V \vdash A_0$ **OK** *and* $A_0 \rightsquigarrow A_1$ *then* $V \vdash A_1$ **OK**.

Proof. By induction on the reduction relation.

- If the last step of the reduction is a congruence rule on a constructor form (RedRedLamA, RedRedLame, RedRedPLamA, RedRedPLame, RedRedRefl) then the safety of the metavariables is proved by the induction hypothesis and the corresponding safety rule (OkOKLam, OkOKPLam, or OkOKRefl).
- RedRedApp1 Then our term is the application $p_1\, p_2$, and its well-formedness is given by one of the two rules:
  - **??** Then the well-formedness follows immediately by the induction hypothesis applied to $V \vdash p_1$ **OK**.
  - **??** Then we will also need to prove $p_1'\, p_2$ **INERT**,

□

4 Normalization

Definition 1. *e is normalized if there is no e' such that $e \rightsquigarrow e'$.*

Definition 2. *e' is a normal form of e if $e \rightsquigarrow^* e'$ and e' is normalized.*

$\boxed{e \textbf{ is neutral}}$

$$\frac{}{x \textbf{ is neutral}} \textsc{RedNeutVar}$$

$$\frac{e_1 \textbf{ is neutral} \quad e_2 \textbf{ is normal}}{e_1\ e_2 \textbf{ is neutral}} \textsc{RedNeutApp}$$

$$\frac{e_1 \textbf{ is neutral} \quad e_2 \textbf{ is normal}}{e_1\{e_2\} \textbf{ is neutral}} \textsc{RedNeutPApp}$$

$$\frac{p \textbf{ is neutral} \quad e_1, e_2, A, B, e \textbf{ are normal}}{\textbf{subst}\ (p : e_1 = e_2 : A, x.\ B, e) \textbf{ is neutral}} \textsc{RedNeutSubst}$$

5 Well-formedness

We define well-formedness of types and terms in the standard congruent way. The interesting part is the base case for the metavariables (WfWFAVar),

$\boxed{\Gamma\ ;\ M\Gamma \vdash e}$

$$\frac{x : A \in \Gamma}{\Gamma\ ;\ M\Gamma \vdash x} \textsc{WfWFVar}$$

$$\frac{\Gamma\ ;\ M\Gamma \vdash A \quad \Gamma, \underline{x} : A\ ;\ M\Gamma \vdash e}{\Gamma\ ;\ M\Gamma \vdash \lambda^{\forall} \underline{x} : A.\ e} \textsc{WfWFPLam}$$

$$\frac{(\Gamma' \vdash \widehat{x} \Leftarrow A) \in M\Gamma \quad \Gamma' \subseteq \Gamma \quad \Gamma\ ;\ M\Gamma \vdash \sigma : \Gamma'}{\Gamma\ ;\ M\Gamma \vdash \widehat{x}[\sigma]} \textsc{WfWFAVar}$$

6 Metavariables and metasubstitution

Definition 3. *Metasubstitution $M\sigma$ is a mapping from metavariables to terms. The specification $\Gamma\ ;\ M\Gamma_2 \vdash M\sigma : M\Gamma_1$ states that*

7 Conversion

Definition 4. *$\Gamma\ ;\ M\Gamma \vdash e_0 \equiv e_1$ is defined as alpha-equivalence transitively closed under reduction.*

8 Unification

Definition 5. *Suppose that two terms e_0 and e_1 are well-typed in the context Γ and metacontext $M\Gamma$, i.e., $\Gamma\ ;\ M\Gamma \vdash e_0$ and $\Gamma\ ;\ M\Gamma \vdash e_1$. Then the metasubstitution $M\sigma_0$ is a unifier of e_0 and e_1 if*

- *$\Gamma\ ;\ M\Gamma_0 \vdash M\sigma_0 : M\Gamma$ and*
- *$\Gamma\ ;\ M\Gamma_0 \vdash [M\sigma_0]e_0 \equiv [M\sigma_0]e_1$.*

The unifier $M\sigma_0$ is the most general if for any other unifier $\Gamma\ ;\ M\Gamma_1 \vdash M\sigma_1 : M\Gamma$ of e_0 and e_1, there exists a metasubstitution $\Gamma\ ;\ M\Gamma_1 \vdash M\tau : M\Gamma_0$ such that $;\ M\Gamma_1 \vdash M\sigma_1 \equiv M\tau \circ M\sigma_0$.

A Appendix

$\boxed{e_1 \rightsquigarrow e_2}$

$$\frac{A_0 \rightsquigarrow A_1}{\lambda^{\Pi} x : A_0.\, e \rightsquigarrow \lambda^{\Pi} x : A_1.\, e} \quad \textsc{RedRedLamA}$$

$$\frac{e_0 \rightsquigarrow e_1}{\lambda^{\Pi} x : A.\, e_0 \rightsquigarrow \lambda^{\Pi} x : A.\, e_1} \quad \textsc{RedRedLamE}$$

$$\frac{A_0 \rightsquigarrow A_1}{\lambda^{\forall} \underline{x} : A_0.\, e \rightsquigarrow \lambda^{\forall} \underline{x} : A_1.\, e} \quad \textsc{RedRedPLamA}$$

$$\frac{e_0 \rightsquigarrow e_1}{\lambda^{\forall} \underline{x} : A.\, e_0 \rightsquigarrow \lambda^{\forall} \underline{x} : A.\, e_1} \quad \textsc{RedRedPLamE}$$

$$\frac{e_0 \rightsquigarrow e_1}{\mathbf{refl}\, e_0 \rightsquigarrow \mathbf{refl}\, e_1} \quad \textsc{RedRedRefl}$$

$$\frac{e_0 \rightsquigarrow e_1}{e_0\, e \rightsquigarrow e_1\, e} \quad \textsc{RedRedApp1}$$

$$\frac{}{(\lambda^{\Pi} x : A.\, e_0)\, e \rightsquigarrow [e/x]e_0} \quad \textsc{RedRedAppLam}$$

$$\frac{}{(\lambda^{\forall} \underline{x} : A.\, e_0)\{e\} \rightsquigarrow [e/\underline{x}]e_0} \quad \textsc{RedRedAppPLam}$$

$$\frac{p_0 \rightsquigarrow p_1}{\mathbf{subst}\,(p_0 : e_1 = e_2 : A, x.\, B, e) \rightsquigarrow \mathbf{subst}\,(p_1 : e_1 = e_2 : A, x.\, B, e)} \quad \textsc{RedRedSubstP}$$

$$\frac{e_0 \rightsquigarrow e_1}{\mathbf{subst}\,(p : e_0 = e_2 : A, x.\, B, e) \rightsquigarrow \mathbf{subst}\,(p : e_1 = e_2 : A, x.\, B, e)} \quad \textsc{RedRedSubstE1}$$

$$\frac{e_0 \rightsquigarrow e_2}{\mathbf{subst}\,(p : e_1 = e_0 : A, x.\, B, e) \rightsquigarrow \mathbf{subst}\,(p : e_1 = e_2 : A, x.\, B, e)} \quad \textsc{RedRedSubstE2}$$

$$\frac{A_0 \rightsquigarrow A}{\mathbf{subst}\,(p : e_1 = e_2 : A_0, x.\, B, e) \rightsquigarrow \mathbf{subst}\,(p : e_1 = e_2 : A, x.\, B, e)} \quad \textsc{RedRedSubstA}$$

$$\frac{B_0 \rightsquigarrow B}{\mathbf{subst}\,(p : e_1 = e_2 : A, x.\, B_0, e) \rightsquigarrow \mathbf{subst}\,(p : e_1 = e_2 : A, x.\, B, e)} \quad \textsc{RedRedSubstB}$$

$$\frac{e_0 \rightsquigarrow e}{\mathbf{subst}\,(p : e_1 = e_2 : A, x.\, B, e_0) \rightsquigarrow \mathbf{subst}\,(p : e_1 = e_2 : A, x.\, B, e)} \quad \textsc{RedRedSubstE}$$

$$\frac{}{\mathbf{subst}\,((\mathbf{refl}\, e_0) : e_1 = e_2 : A, x.\, B, e) \rightsquigarrow e} \quad \textsc{RedRedSubstRed}$$

$\boxed{\Gamma\,;M\Gamma \vdash e}$

$$\frac{x : A \in \Gamma}{\Gamma\,;M\Gamma \vdash x} \quad \textsc{WfWFVar}$$

$$\frac{}{\Gamma\,;M\Gamma \vdash <>}\ \textsc{WfWFUnit}$$

$$\frac{\Gamma\,;M\Gamma \vdash A \quad \Gamma, x : A\,;M\Gamma \vdash e}{\Gamma\,;M\Gamma \vdash \lambda^{\Pi} x : A.\, e}\ \textsc{WfWFLam}$$

$$\frac{\Gamma\,;M\Gamma \vdash A \quad \Gamma, \underline{x} : A\,;M\Gamma \vdash e}{\Gamma\,;M\Gamma \vdash \lambda^{\forall} \underline{x} : A.\, e}\ \textsc{WfWFPLam}$$

$$\frac{\Gamma\,;M\Gamma \vdash e_1 \quad \Gamma\,;M\Gamma \vdash e_2}{\Gamma\,;M\Gamma \vdash e_1\, e_2}\ \textsc{WfWFApp}$$

$$\frac{\Gamma\,;M\Gamma \vdash e_1 \quad \Gamma\,;M\Gamma \vdash e_2}{\Gamma\,;M\Gamma \vdash e_1\{e_2\}}\ \textsc{WfWFPApp}$$

$$\frac{\Gamma\,;M\Gamma \vdash e_1}{\Gamma\,;M\Gamma \vdash e_1\{\_\}}\ \textsc{WfWFPAppU}$$

$$\frac{\Gamma\,;M\Gamma \vdash e}{\Gamma\,;M\Gamma \vdash \mathbf{refl}\, e}\ \textsc{WfWFRefl}$$

$$\frac{\begin{array}{c}\Gamma\,;M\Gamma \vdash p \quad \Gamma\,;M\Gamma \vdash e_1 \quad \Gamma\,;M\Gamma \vdash e_2 \\ \Gamma\,;M\Gamma \vdash A \quad \Gamma, x : A\,;M\Gamma \vdash B \quad \Gamma\,;M\Gamma \vdash e\end{array}}{\Gamma\,;M\Gamma \vdash \mathbf{subst}\,(p : e_1 = e_2 : A, x.\, B, e)}\ \textsc{WfWFSubst}$$

$$\frac{\begin{array}{c}\Gamma\,;M\Gamma \vdash e_1 \quad \Gamma\,;M\Gamma \vdash A \\ \Gamma, x : A\,;M\Gamma \vdash e_2\end{array}}{\Gamma\,;M\Gamma \vdash \mathbf{let}\, x : A = e_1\, \mathbf{in}\, e_2}\ \textsc{WfWFLet}$$

$$\frac{(\Gamma' \vdash \widehat{x} \Leftarrow A) \in M\Gamma \quad \Gamma' \subseteq \Gamma \quad \Gamma\,;M\Gamma \vdash \sigma : \Gamma'}{\Gamma\,;M\Gamma \vdash \widehat{x}[\sigma]}\ \textsc{WfWFAVar}$$