# 1 Syntax

**Definition 1** (Syntax of terms)**.** *We write term meaning a computation or a value, denoting it $\sigma$ or $\tau$.*

$$\begin{array}{llll} \textit{Computations} & X, Y, t, u & ::= & t v \mid \uparrow A \mid \Pi x : A .\, X \mid \forall x : A .\, X \mid \mathsf{force}\, v \mid \mathsf{return}\, v \mid \lambda x : A .\, t \mid \\ & & & \mathsf{let}\ x : A := t\ \mathsf{in}\ u \mid \mathsf{dlet}\ x : A := t\ \mathsf{in}\ u \mid \mathsf{rec}_{\mathsf{eq}}^{x.y.X}(v, t) \\ & & & \\ \textit{Values} & A, B, v, w & ::= & x \mid a \mid \downarrow X \mid \mathsf{refl}_v \mid \mathsf{eq} A\, v\, w \mid \{t\} \end{array}$$

We use different non-terminal symbols to emphasize the distinction between type-level terms and term-level terms, which manifests properly in section 9. The upper-case literals represent type-terms, and the lower-case represent term-terms (which can be typed with some type-terms) with one exception: in $\mathsf{let}\ x : A := t\ \mathsf{in}\ u$, $u$ can represent a type-term.

We reserve $x$ and $y$ to denote term-level variables, and $a$ and $b$ to denote type-level variables.

# 2 Computational form of the terms

Let us consider the term syntax from a different perspective:

**Definition 2** (Computational syntax of terms)**.**

$$\begin{array}{llll} \textit{Constructors} & C & ::= & \lambda \mid \hat{\lambda} \mid \mathsf{refl} \mid \mathsf{return} \mid \{\} \\ & & & \\ \textit{Eliminators} & E & ::= & @ \mid \mathsf{rec}_{\mathsf{eq}}(,) \mid \mathsf{let} \mid \mathsf{dlet} \mid \mathsf{force} \\ & & & \\ \textit{Neutral Formers} & N & ::= & \downarrow \mid \Pi \mid \forall \mid \uparrow \mid x \\ & & & \\ \textit{Formers} & F & ::= & C \mid E \mid N \\ & & & \\ \textit{Abstractor Heads} & \vec{x}^{0} & ::= & . \\ & \vec{x}^{n+1} & ::= & x \,.\, \vec{x}^{n} \\ & \vec{x} & ::= & \vec{x}^{0} \mid \vec{x}^{1} \mid \ldots \\ & & & \\ \textit{Abstractors} & P^n, Q^n & ::= & \vec{x}^{n}\tau \\ & P, Q & ::= & P^1 \mid P^2 \mid \ldots \\ & & & \\ \textit{Terms} & \sigma, \tau, \nu & ::= & F\,(\vec{x}^{\mathsf{ar}F_1}\tau_1)\ldots(\vec{x}^{\mathsf{ar}F_{|F|}}\tau_{|F|}) \end{array}$$

**Definition 3** (Arity)**.** *For every term former $F$ we define its arity $\mathsf{ar}F$ as the array of integers desctibing its arguments. Integer denotes the number of new binding variables "created" by $F$ that can be used in the corresponding subterm. For brevity, we denote length of $\mathsf{ar}F$ as $|F|$.*

| $F$ | $\lambda$ | $\hat{\lambda}$ | $(,)$ | refl | return | $\{\}$ | $@$ | $\mathsf{rec}_{\mathsf{eq}}$ |
|---|---|---|---|---|---|---|---|---|
| $\mathsf{ar}F$ | $[0,1]$ | $[0,1]$ | $[0,0]$ | $[]$ | $[0]$ | $[0]$ | $[0,0]$ | $[0,2,0]$ |
| $\|F\|$ | 2 | 2 | 2 | 0 | 1 | 1 | 2 | 3 |

| $F$ | let | dlet | force | $\downarrow$ | $\Pi$ | $\forall$ | $\uparrow$ | $x$ |
|---|---|---|---|---|---|---|---|---|
| $\mathsf{ar}F$ | $[0,1,0]$ | $[0,1,0]$ | $[0]$ | $[0]$ | $[0,1]$ | $[0,1]$ | $[0]$ | $[]$ |
| $\|F\|$ | 3 | 3 | 1 | 1 | 2 | 2 | 1 | 0 |

Ilya: Notice that we rearrange the arguments in $\lambda$ and let-bindings so that any redex is always an eliminator whose first argument is a constructor

It is easy to see that the syntax of terms from definition 1 defines the subset of terms defined by definition 2. In fact, any well-typed term must have a form defined by definition 1. We will use these two representation interchangeably.

## 3 Alpha-equivalence

**Definition 4** (Variable Renaming)**.**

$$\frac{}{x\{x \rightsquigarrow z\} = z} \qquad \frac{x \neq y}{x\{y \rightsquigarrow z\} = x} \qquad \frac{F \neq x}{F\,P_1 \dots P_{|F|}\{y \rightsquigarrow z\} = F\,(P_1\{y \rightsquigarrow z\}) \dots (P_{|F|}\{y \rightsquigarrow z\})}$$

$$\frac{}{.\tau\{y \rightsquigarrow z\} = .(\tau\{y \rightsquigarrow z\})} \qquad \frac{x' \textit{ is fresh}}{x.P\{y \rightsquigarrow z\} = x'.((P\{x \rightsquigarrow x'\})\{y \rightsquigarrow z\})}$$

**Definition 5** (Alpha-equivalence)**.**

$$\frac{\forall i, P_i \sim_\alpha Q_i}{F\,P_1 \dots P_{|F|} \sim_\alpha F\,Q_1 \dots Q_{|F|}} \qquad \frac{\sigma \sim_\alpha \tau}{.\sigma \sim_\alpha .\tau} \qquad \frac{y \textit{ is fresh} \qquad \sigma\{x \rightsquigarrow y\} \sim_\alpha \tau\{x \rightsquigarrow y\}}{x.\sigma \sim_\alpha x.\tau}$$

**Lemma 1.** *Alpha-equivalence is an equivalence relation on terms and abstractors.*

Ilya: Admitted.

**Lemma 2** (Functionality of Variable Renaming)**.** *Variable Renaming is a functional on the classes of alpha-equivalence.*

Ilya: Admitted.

Hereafter, we assume every statement about terms and abstractors defined on the equivalence classes. Whenever we use the "concrete term syntax", we mean the alpha-equivalence class of this term if the term is in the covariant position of the statement or definition (e.g. we are constructing a function returning an equivalence class as an output); and any term of this form from this class if the term is in the contravariant position (e.g. we are constructing a function taking an equivalence class as an input).

## 4 Susbtititution

**Definition 6** (Substitution)**.** *Ilya: todo*

**Lemma 3** (Functionality of Substituion)**.** *Substitution is a functional on the classes of alpha-equivalence.*

Ilya: Admitted.

## 5 Reduction

First, we define the redex contraction.

**Definition 7** (Redex Contraction)**.** *We define the top-level redex contraction in the following way:*

- $(\lambda x : \nu . \sigma)\tau \rightharpoonup \sigma\{x := \tau\}$
- $\mathsf{let}\ x : \nu\ := \mathsf{return}\,\sigma\ \mathsf{in}\ \tau \rightharpoonup \tau\{x := \sigma\}$
- $\mathsf{dlet}\ x : \nu\ := \mathsf{return}\,\sigma\ \mathsf{in}\ \tau \rightharpoonup \tau\{x := \sigma\}$
- $\mathsf{force}\,\{\tau\} \rightharpoonup \tau$
- $\mathsf{rec}^{\nu}_{\mathsf{eq}}(\mathsf{refl}_v, \tau) \rightharpoonup \tau$

*The terms on the left hand side of $\cdot \rightharpoonup \cdot$ are called redexes.*

Notice that any redex from definition 7 is an elimination of a constructor, i.e. a term of the form $E\,(C\,P_1 \ldots P_{|C|})\,Q_2 \ldots Q_{|E|}$ where $E$ and $C$ are "matched". Vice versa, if a term of the form $E\,(C\,P_1 \ldots P_{|C|})\,Q_2 \ldots Q_{|E|}$ is well-typed, it is a redex.

Informally, reduction of a term $\tau$ is a redex contraction happening in some subterm of $\tau$.

**Definition 8** (Reduction).

$$\frac{\tau \rightharpoonup \tau'}{\tau \to \tau'}\ \textsc{Redex} \qquad\qquad \frac{\tau \to \tau'}{F\,P_1 \ldots (\vec{x}^{\,\mathsf{ar}F_i}\tau) \ldots P_{|F|} \to F\,P_1 \ldots (\vec{x}^{\,\mathsf{ar}F_i}\tau') \ldots P_{|F|}}\ \textsc{Cong}_i^F$$

**Lemma 4** (Substitution preserves reduction).

$$\frac{\tau \to \tau'}{\tau\{x := \sigma\} \to \tau'\{x := \sigma\}}$$

*Proof.* Induction on $\tau \to \tau'$. Substitution is congruent, therefore, the induction goes down to the redexes.

- Suppose that $(\lambda x : \nu.\, \sigma)\sigma' \to \sigma\{x := \sigma'\}$. We need to prove that $(\lambda x : \nu.\, \sigma)\sigma'\{y := \tau\} \to \sigma\{x := \sigma'\}\{y := \tau\}$. We know that $(\lambda x : \nu.\, \sigma)\sigma'\{y := \tau\} = (\lambda x : \nu.\, \sigma\{y := \tau\})(\sigma'\{y := \tau\})$, which reduces to $\sigma\{y := \tau\}\{x := \sigma'\{y := \tau\}\}$. But
$$\sigma\{y := \tau\}\{x := \sigma'\{y := \tau\}\} = \sigma\{x := \sigma'\}\{y := \tau\},$$
assuming that $x \notin \mathsf{FV}(\tau)$, which is guaranteed because the substitution is capture-avoiding.
- The other cases are similar or straightforward

□

## 6 Normal Form

Using the syntax from definition 2, it is convenient to express computational properties of the term, e.g. being in the normal form ($\mathsf{NF}$).

**Definition 9** (Normal Form).

$$\frac{\tau\ \mathsf{ATOM}}{\tau\ \mathsf{NF}} \qquad \frac{\tau_1\ \mathsf{NF}\ \ldots\ \tau_{|C|}\ \mathsf{NF}}{C\ \vec{x}\tau_1 \ldots \vec{x}\tau_{|C|}\ \mathsf{NF}} \qquad \frac{\tau_1\ \mathsf{NF}\ \ldots\ \tau_{|N|}\ \mathsf{NF}}{N\ \vec{x}\tau_1 \ldots \vec{x}\tau_{|N|}\ \mathsf{ATOM}} \qquad \frac{\tau_1\ \mathsf{ATOM} \qquad \tau_2\ \mathsf{NF}\ \ldots\ \tau_{|E|}\ \mathsf{NF}}{E\ \vec{x}\tau_1 \ldots \vec{x}\tau_{|E|}\ \mathsf{ATOM}}$$

The intuition is that (i) normal terms are not reducible; (ii) atomic terms are not reducible and, in addition, do not cause reduction when the eliminators are applied to them.

Although it is easy to see that the terms in normal form are not reducible, the opposite is only true for the well-typed terms:

**Proposition 1** (Normal form and irreducibility).

$$\frac{\tau\ \mathsf{NF}}{\nexists \tau', \tau \to \tau'} \qquad\qquad \frac{\tau\ \textit{is well-typed} \qquad \nexists \tau', \tau \to \tau'}{\tau\ \mathsf{NF}}$$

**Definition 10** (Reduction to the Normal Form).

$$\frac{\tau\ \mathsf{NF}}{\tau \Downarrow \tau} \qquad\qquad \frac{\tau \to \tau' \qquad \tau' \Downarrow \tau''}{\tau \Downarrow \tau''}$$

**Proposition 2** (Reduction-Substitution distributivity).

$$\Downarrow(\sigma\{x := \Downarrow v\}) = \Downarrow(\sigma\{x := v\})$$

# 7 Safe Occurrence

Ilya: Hmm.. Discussing the type system before actually introducing it...

In this section, we describe the basis of the restriction that we put on the type system to make the type checking and type inference decidable. First, let us motivate why we need to restrict the type system.

## 7.1 The necessity of the restrictions

We motivate the restrictions that we put on the type system by considering two well-known undecidable problems: the Inhabitation problem and Hilbert's tenth problem. We will show that both of them can be reduced to the typing problems in the unrestricted system.

**The Inhabitation** Let us consider the term $\langle\rangle$ checked against type $\forall x : \tau.\, \mathsf{1}$. It is easy to see that the checking succeed if and only if $\tau$ is inhabited. Thus, to ensure the decidability of the type checking, we require $x$, bound by some $\forall$, to occur within the body of that $\forall$. In other words, only do we allow $\forall x : \sigma.\, \tau$ to be formed, if $x \in \mathsf{FV}(\tau)$.

**Hilbert's Tenth Problem** As will be shown in sections 9, 9.1 and 10, the "driver" of the type inference algorithm is the subtyping algorithm, and the "driver" of the subtyping, is the unification. Let us show that in the unrestricted case, the unification is undecidable.

The type system can be easily extended with natural numbers. To this purpose, we must add $\mathbb{N}$, $\mathsf{0}$, and $\mathsf{succ}(v)$ to the values, and $\mathsf{rec}^X_{\mathbb{N}}(v, \mathit{base}, \mathit{step})$ to the computations with obvious typing inference rules. We also add $\mathsf{rec}^X_{\mathbb{N}}(\mathsf{0}, b, s) \rightharpoonup b$ and $\mathsf{rec}^X_{\mathbb{N}}(\mathsf{succ}(v), b, s) \rightharpoonup s\, v\, \mathsf{rec}^X_{\mathbb{N}}(v, b, s)$ to the contraction rules.

Then the unification that we will not provide is, for example, the following: $\mathsf{rec}^X_{\mathbb{N}}(\hat{v}, \mathsf{0}, \lambda x\, y.\, \mathsf{0}) \equiv \mathsf{0}$, where $\hat{v}$ is the unification variable that we must initialize. Although $(\hat{v}{:=}\mathsf{0})$ solves this unification, we rule out such cases from our unification algorithm. This is because having the unification variable on the reducible position means being able to "invert" the recursion, which is too powerful to be decidable.

Specifically, after we defined integers, arithmetic operations, we can define any arbitrary polynomial $P(\hat{x}_1, \dots, \hat{x}_n)$. The unification of this polynomial with $\mathsf{0}$ corresponds to solving a diophantine equation, which is undecidable.

## 7.2 The Restriction on Quantifiers

To deal with both aforementioned undecidabilities, we introduce a syntactic judgement "safe occurrence of the variable". The judgement $x \in^{?} \tau\, \mathsf{OK}$ means $x$ occurs safely in $\tau$.

Ideally, we would like to forbid the situations when in some normal form of $\tau$, some instantiation of $x$ generates a new redex. In other words, we would like to ensure that all normal forms of $\tau$ do not contain $E\, x\, \tau_2 \dots \tau_{|E|}$ as a subterm.

However, this property is undecidable by Rice's theorem. Notice that (i) we do not require terms to have types at this stage, thus, the system is Turing complete; (ii) the property is non-trivial; (iii) the property judges about the normal forms and thus, is invariant under "algorithmic equivalence".

As it is undecidable, it is impossible to express this judgement using well-founded inference rules (i.e. unambiguously generating finite trees). Since precise syntactic representation of this property is impossible, we under-approximate this property via $x \in^{?} \tau\, \mathsf{OK}$ judgement:

**Definition 11** (Safe Occurrence)**.** *Ilya: TODO: add safe occurrence in abstractors*

$$\frac{x \in^? \tau_1\,\mathsf{OK}\ \ \ldots\ \ x \in^? \tau_{|C|}\,\mathsf{OK}}{x \in^? C\,\vec{x}\tau_1 \ldots \vec{x}\tau_{|C|}\,\mathsf{OK}}\ \textsc{C-Cong} \qquad\qquad \frac{x \in^? \tau_1\,\mathsf{OK}\ \ \ldots\ \ x \in^? \tau_{|N|}\,\mathsf{OK}}{x \in^? N\,\vec{x}\tau_1 \ldots \vec{x}\tau_{|N|}\,\mathsf{OK}}\ \textsc{N-Cong}$$

$$\frac{x \notin \mathsf{FV}(E\,\tau_1 \ldots \tau_{|E|})}{x \in^? E\,\vec{x}\tau_1 \ldots \vec{x}\tau_{|E|}\,\mathsf{OK}}\ \text{E-FV}$$

$$\frac{x \in^? \tau_1\,\mathsf{OK}\ \ \ldots\ \ x \in^? \tau_{|E|}\,\mathsf{OK} \qquad \tau_1 \neq x \qquad E\,\vec{x}\tau_1 \ldots \vec{x}\tau_{|E|}\,\mathsf{INERT}}{x \in^? E\,\vec{x}\tau_1 \ldots \vec{x}\tau_{|E|}\,\mathsf{OK} \qquad \textit{Ilya: (implicit } \alpha\textit{-rename!)}}\ \textsc{E-Cong}$$

In the last rule, "$\tau_1 \neq x$" means literal syntactic inequality. Intuitively, "$\tau\,\mathsf{INERT}$" means that $\tau$ preserves its top-level structure under the reduction, i.e. the reduction always happens in the subterms of $\tau$ but never on the top level. In fact, the relation we define is a little bit stronger, as it also forbids changing of the structure of the eliminator's first argument. Formally, it is defined as follows:

**Definition 12** (Inert Terms)**.**

$$\frac{}{N\,\vec{x}\tau_1 \ldots \vec{x}\tau_{|N|}\,\mathsf{INERT}} \qquad \frac{}{C\,\tau_1 \ldots \tau_{|C|}\,\mathsf{INERT}} \qquad \frac{}{E\,(N\,\vec{x}\sigma_1 \ldots \vec{x}\sigma_{|N|})\,\vec{x}\tau_2 \ldots \vec{x}\tau_{|E|}\,\mathsf{INERT}}$$

$$\frac{E'\,\vec{x}\sigma_1 \ldots \vec{x}\sigma_{|E'|}\,\mathsf{INERT}}{E\,(E'\,\vec{x}\sigma_1 \ldots \vec{x}\sigma_{|E'|})\,\vec{x}\tau_2 \ldots \vec{x}\tau_{|E|}\,\mathsf{INERT}}\ \textsc{EE-Inert}$$

As a heuristics, it is possible to extend the "Safe Occurrence" property by embedding some of the redex contractions from definition 7 into the inference system. Notice that only non-substituting contractions are allowed. This is because otherwise, we embed the full evaluation into the inference system, which makes it non-well-founded (and undecidable).

**Definition 13** (Safe Occurrence Extension)**.**

$$\frac{x \in^? \tau\,\mathsf{OK}}{x \in^? \mathsf{force}\,\{\tau\}\,\mathsf{OK}} \qquad \frac{x \in^? @\,(@\,\sigma\,\tau_1)\,\tau_2\,\mathsf{OK} \qquad x \notin \tau'}{x \in^? \mathsf{rec}^{\tau'}_{\Sigma}(\langle \tau_1, \tau_2 \rangle, \sigma)\,\mathsf{OK}} \qquad \frac{x \notin \sigma x \in^? \tau\,\mathsf{OK} \qquad x \in^? \upsilon\,\mathsf{OK}}{x \in^? \mathsf{rec}^{x.y.\sigma}_{\mathsf{eq}}(\mathsf{refl}_{\upsilon}, \tau)\,\mathsf{OK}}$$

*Ilya: The blue rules are experimental!*

$$\frac{x \notin \tau \qquad x \notin \nu \qquad x \in^? \sigma\,\mathsf{OK}}{x \in^? (\lambda y : \nu.\,\sigma)\tau\,\mathsf{OK}} \qquad\qquad \frac{x \notin \sigma \qquad x \notin \nu \qquad x \in^? \tau\,\mathsf{OK}}{x \in^? \mathsf{let}\ y : \nu\ :=\ \mathsf{return}\,\sigma\ \mathsf{in}\ \tau\,\mathsf{OK}}$$

$$\frac{x \notin \sigma \qquad x \notin \nu \qquad x \in^? \tau\,\mathsf{OK}}{x \in^? \mathsf{dlet}\ y : \nu\ :=\ \mathsf{return}\,\sigma\ \mathsf{in}\ \tau\,\mathsf{OK}}$$

**Lemma 5** (Congruence of the safe occurrence)**.**

$$\frac{x \in^? F\,P_1 \ldots P_{|F|}\,\mathsf{OK}}{x \in^? P_1\,\mathsf{OK} \qquad \cdots \qquad x \in^? P_{|F|}\,\mathsf{OK}}$$

*Proof.* Trivial induction. □

**Lemma 6** (Monadicity of the safe substitution)**.**

$$\frac{x, y \in^? \sigma\,\mathsf{OK} \qquad y \in^? \tau\,\mathsf{OK}}{y \in^? \sigma\{x\ :=\ \tau\}\,\mathsf{OK}}$$

**Lemma 7** (Reduction-Substitution Commutativity)**.**

$$\frac{x \in^? \sigma\,\mathsf{OK} \qquad \tau\,\mathsf{NF} \qquad \sigma\{x := \tau\} \to \sigma'}{\exists \sigma^* \ s.t.\ \sigma \to \sigma^* \ and\ \sigma^*\{x := \tau\} = \sigma'}$$

*Or in the commutative diagram form: if $x \in^? \sigma\,\mathsf{OK}$ and $\tau\,\mathsf{NF}$ then*

$$\begin{tikzcd} \sigma \arrow[r, dashed, "\to"] \arrow[d, "x:=\tau"'] & \sigma^* \arrow[d, dashed, "x:=\tau"] \\ \bullet \arrow[r, "\to"'] & \sigma' \end{tikzcd}$$

*Proof.* Let us destruct the substitution $\sigma\{x := \tau\}$. Notice that $\sigma \neq x$ because $x\{x := \tau\} = \tau \not\twoheadrightarrow \cdot$. It means that the substitution is performed by congruence: $\sigma = F\,\sigma_1 \ldots \sigma_{|F|}$ (for some $F \neq x$), and $\sigma\{x := \tau\} = F\,(\sigma_1\{x := \tau\}) \ldots (\sigma_{|F|}\{x := \tau\})$. Notice that $x \in^? \sigma_i\,\mathsf{OK}$ for $i = 1 \ldots |F|$ by lemma 5.

Induction on $\sigma\{x := \tau\} \to \sigma'$. The reduction step can be justified either by the congruence or the redex contraction.

- If the reduction step is done by congruence, then the required $\sigma^*$ is of the form $F\,\sigma_1^* \ldots \sigma_{|F|}^*$ where $\sigma_1^* \ldots \sigma_{|F|}^*$ are constructed by the straightforward application of the induction hypothesis to $\sigma_1 \ldots \sigma_{|F|}$.
- If the reduction is the top-level redex contraction, then $\sigma\{x := \tau\}$ is a redex, i.e. $F$ is an eliminator $E$ and $\sigma_1\{x := \tau\}$ is formed by a constructor $C$. Notice that because $x \in^? E\,\sigma_1 \ldots \sigma_{|E|}\,\mathsf{OK}$, $\sigma_1 \neq x$. Therefore, the substitution $\sigma_1\{x := \tau\}$ is also done by congruence: $\sigma_1 = C\,\zeta_1 \ldots \zeta_{|C|}$ and thus, $\sigma = E\,(C\,\zeta_1 \ldots \zeta_{|C|})\,\sigma_2 \ldots \sigma_{|E|}$.

  Let us destruct $x \in^? \sigma\,\mathsf{OK}$. Since $\sigma$ is not inert, either (i) $x \notin \mathsf{FV}(\sigma)$, then the substitution is the identity, and we can take $\sigma^* = \sigma'$); or (ii) one of the "additional" rules is applied to get $x \in^? \sigma\,\mathsf{OK}$. In all of these three cases, we can perform the same top-level redex contraction to acquire $\sigma^*$. This operation commutes with substitution because all it does is restructuring the top-level form of $\sigma$ without changing the subterms $\zeta_1, \ldots, \zeta_{|C|}, \sigma_2, \ldots, \sigma_{|E|}$, thus, the required property holds. Ilya: to be fair, the beta-reduction also commutes with the substitution, but we still need the inertness so that OK is preserved under reduction.

□

**Corollary 1** (Normalization-Substitution Commutativity)**.**

$$\frac{x \in^? \sigma\,\mathsf{OK} \qquad \tau\,\mathsf{NF}}{\Downarrow(\sigma\{x := \tau\}) = (\Downarrow\sigma)\{x := \tau\}} \qquad\qquad \frac{x \in^? \sigma\,\mathsf{OK}}{\Downarrow(\sigma\{x := \tau\}) = (\Downarrow\sigma)\{x := \Downarrow\tau\}}$$

*Proof.* By replicating lemma 7. If $\tau$ is not in the normal form, proposition 2 is applied. □

**Lemma 8** (Reduction preserves inertness)**.**

$$\frac{\tau\,\mathsf{INERT} \qquad \tau \to \tau'}{\tau'\,\mathsf{INERT}}$$

*Proof.* Induction on $\tau\,\mathsf{INERT}$. □

**Lemma 9** (Reduction preserves safe occurrence)**.**

$$\frac{x \in^? \tau\,\mathsf{OK} \qquad \tau \to \tau'}{x \in^? \tau'\,\mathsf{OK}}$$

*Proof.* Induction on $x \in^? \tau\,\mathsf{OK}$.

- For C-Cong (N-Cong), we apply the induction hypothesis and C-Cong (N-Cong, resp.).
- For E-FV, notice that the reduction does not increase the set of free variables, and thus, E-FV is applicable after the reduction of one of the $\tau_i$.
- The E-Cong case is a little bit more complicated. Notice that $\tau_1 \nrightarrow x$. This is because if $\tau_1$ is an eliminator, it must be inert by EE-Inert. Then we can consider in which $\tau_i$ the reduction happened, apply the induction hypothesis and lemma 8.
- For the additional rules, the reduction can be either by congruence (and then we apply the induction hypothesis, lemma 5 and the same rule) or by the top-level redex contraction, and then the required property is exactly one of the premises.

□

Notice that no occurrence means safe occurrence, that is $x \in^? \tau\,\mathsf{OK}$ does not imply $x \in \mathsf{FV}(\tau)$. We use notation $x \in^! \tau\,\mathsf{OK}$ to denote $x \in^? \tau\,\mathsf{OK}$ and $x \in \mathsf{FV}(\tau)$.

**Definition 14** (Strictly Safe Occurrence).

$$\frac{x \in^? \tau\,\mathsf{OK} \qquad x \in \mathsf{FV}(\tau)}{x \in^! \tau\,\mathsf{OK}}$$

**Proposition 3** (Safe occurrence in a redex). *If $\tau$ is a top-level redex and $x \in^! \tau\,\mathsf{OK}$, then the redex is non-substitutive, i.e. the judgement $x \in^? \tau\,\mathsf{OK}$ was constructed by one of the rules from $definition$* 13

**Lemma 10** (Reduction preserves strictly safe occurrence).

$$\frac{x \in^! \tau\,\mathsf{OK} \qquad \tau \to \tau'}{x \in^! \tau'\,\mathsf{OK}}$$

*Proof.* Induction on $\tau \to \tau'$. Since the redex contraction is special case of reduction, by lemma 9, it preserves the non-strict safe occurrence. Let us show that the actual occurrence $x \in \mathsf{FV}(\tau)$ is preserved as well.

$x$ can only disappear after the contraction of the redex $r \subseteq \tau$ if $x$ occurred inside this redex (i.e. $x \in \mathsf{FV}(r)$). By the trivial implication of lemma 5, since $r \subseteq \tau$ and $x \in^? \tau\,\mathsf{OK}$, $x \in^? r\,\mathsf{OK}$. Then by proposition 3, $r$ can only be non-substitutive. However, non-substitutive redexes preserve the variables, hence, $x$ could not have disappeared.

□

**Definition 15** (Safe Existential Variable Set).

$$\mathsf{EV}^{\mathsf{OK}}(\rho) = \{\hat{x} \mid \hat{x} \in^! \rho\,\mathsf{OK}\}$$

**Corollary 2** (Reduction preserves the Safe Existential Variable Set).

$$\frac{\tau \to \tau'}{\mathsf{EV}^{\mathsf{OK}}(\tau) = \mathsf{EV}^{\mathsf{OK}}(\tau')} \qquad \frac{\tau \Downarrow \tau'}{\mathsf{EV}^{\mathsf{OK}}(\tau) = \mathsf{EV}^{\mathsf{OK}}(\tau')}$$

**Definition 16** (Existential Variable Set).

$$\mathsf{EV}(\rho) = \{\hat{x} \mid \hat{x} \in \mathsf{FV}(\rho)\}$$

**Proposition 4** (Existential Variable Set is monotonous w.r.t. Reduction).

$$\frac{\tau \to \tau'}{\mathsf{EV}(\tau) \supseteq \mathsf{EV}(\tau')} \qquad \frac{\tau \Downarrow \tau'}{\mathsf{EV}(\tau) \supseteq \mathsf{EV}(\tau')}$$

# 8 Equivalence and Unification

**Definition 17** (Syntax of algorithmic terms). *Throughout the algorithm, we will use the auxiliary pre-cooked terms, containing some unassigned parts. For this purpose, we extend the syntax of terms (definition 1) by adding the "hatted" unification (existential) variables $\hat{x}$ to the set of values:*

$$\textit{Values} \ \mathrel{+}= \ \hat{x}$$

*Similarly, we extend the syntax from definition 2 by adding $\hat{x}$ to the Neutral Formers:*

$$\textit{Neutral Formers} \ \mathrel{+}= \ \hat{x}$$

**Notation 1.** *To denote that the term is algorithmic, i.e. potentially contains the unification variables, we use $\pi$ and $\rho$. If the term does not contain the unification variables it is called ground and denoted as $\sigma$ and $\tau$.*

**Definition 18** (Safe algorithmic term)**.** *We say that the algorithmic term $\rho$ is safe iff all the unification variables occur safely in it:*

$$\frac{\forall \hat{x},\ \hat{x} \in^{?} \rho\,\mathsf{OK}}{\rho\,\mathsf{OK}}$$

**Definition 19** (Untyped Binding Context)**.**

$$\gamma ::= \cdot \ \mid \ \gamma,\ x \ \mid \ \gamma,\ \hat{x}$$

**Definition 20** (Equivalence)**.** *We define equivalence on terms. Notice that the terms are not necessarily ground. However, the unification variables are treated as the normal ones.*

**Reduction closure** Ilya: We can pack it in one rule using $\Downarrow$

$$\frac{\rho_1 \to \rho_1' \qquad \gamma \vdash \rho_1' \equiv \rho_2}{\gamma \vdash \rho_1 \equiv \rho_2}\ \textsc{Red-L} \qquad\qquad \frac{\rho_2 \to \rho_2' \qquad \gamma \vdash \rho_1 \equiv \rho_2' \qquad \rho_1\,\mathsf{NF}}{\gamma \vdash \rho_1 \equiv \rho_2}$$

**Congruence**

$$\frac{\gamma \vdash P_1 \equiv Q_1 \qquad \ldots \qquad \gamma \vdash P_{|F|} \equiv Q_{|F|} \qquad F\,P_1 \ldots P_{|F|}\,\mathsf{NF} \qquad F\,Q_1 \ldots Q_{|F|}\,\mathsf{NF}}{\gamma \vdash F\,P_1 \ldots P_{|F|} \equiv F\,Q_1 \ldots Q_{|F|}}$$

$$\frac{\gamma, x \vdash \overrightarrow{x}^{n}\pi \equiv \overrightarrow{x}^{n}\rho}{\gamma \vdash x.\overrightarrow{x}^{n}\pi \equiv x.\overrightarrow{x}^{n}\rho} \qquad\qquad \frac{\gamma \vdash \pi \equiv \rho}{\gamma \vdash .\pi \equiv .\rho}$$

**Lemma 11** (Equivalent terms have equal normal forms)**.**

$$\frac{\Gamma \vdash \pi \equiv \rho}{\Downarrow\pi = \Downarrow\rho}$$

**Lemma 12** (Equivalent terms have equal safe existential variable sets)**.**

$$\frac{\Gamma \vdash \pi \equiv \rho}{\mathsf{EV}^{\mathsf{OK}}(\pi) = \mathsf{EV}^{\mathsf{OK}}(\rho)}$$

*Proof.* Trivial induction using corollary 2. □

**Definition 21** (Unification Context)**.** *Unification context represents a (partial) solution of the unification problem.*

$$\theta ::= \cdot \ \mid \ \theta,\ x \ \mid \ \theta,\ \hat{x} \ \mid \ \theta,\ (\hat{x}{:=}\tau)$$

*where $\tau$ is a ground term.*

Here the existential variables $\hat{x}$ play a role different to the one of normal variables $x$. They denote the values that need to be resolved.

**Definition 22** (Variables of the Unification Context)**.**

$$\mathsf{vars}(\cdot) = \varnothing \qquad \mathsf{vars}(\theta,\ \hat{x}) = \mathsf{vars}(\theta) \cup \{\hat{x}\}$$

$$\mathsf{vars}(\theta,\ x) = \mathsf{vars}(\theta) \cup \{x\} \qquad \mathsf{vars}(\theta,\ (\hat{x}{:=}\cdot)) = \mathsf{vars}(\theta) \cup \{\hat{x}\}$$

**Definition 23** (Admissible Terms)**.**

$$\frac{\mathsf{FV}(\rho) \subseteq \mathsf{vars}(\theta)}{\theta \vdash \rho}$$

**Definition 24** (Well-Formed Unification Context)**.** *The unification context $\theta$ is well-formed if every unification variable occurs in it only once, and if it is solved (i.e. the entry $(\hat{x}{:=}\tau)$) occurs in $\theta$, the solution term $\tau$ is in the normal form and is admissible by the prefix of this entry.*

$$\frac{}{\vdash\,\cdot}\qquad\frac{\vdash\theta\qquad x\notin\theta}{\vdash\theta, x}\qquad\frac{\vdash\theta\qquad\hat{x}\notin\theta\qquad(\hat{x}{:=}\,\cdot\,)\notin\theta}{\vdash\theta,\hat{x}}\qquad\frac{\vdash\theta\qquad\hat{x}\notin\theta\qquad\tau\,\mathsf{NF}\qquad\theta\vdash\tau}{\vdash\theta,(\hat{x}{:=}\tau)}$$

**Definition 25** (Application of the Well-Formed Context)**.** *If the unification context $\theta$ is well-formed, We write $[\theta]\rho$ meaning the application of the partial (substitution) function represented by $\theta$ to the term $\rho$:*

$$[\,\cdot\,]\rho = \rho \qquad\qquad [\theta,\,\hat{x}]\rho = \rho$$

$$[\theta,\,x]\rho = \rho \qquad\qquad [\theta,\,(\hat{x}{:=}\sigma)]\rho = ([\theta]\rho)\{\hat{x}\ :=\ \sigma\}$$

Intuitively, when a context is applied to a term, the components of the context are applied to the term one-by-one. This way, the properties holding for a single substitution, can be lifted up to the context application.

**Corollary 3** (Context application commutes with the reduction)**.**

$$\frac{\vdash\theta\qquad\rho\,\mathsf{OK}\qquad[\theta]\rho\to\rho'}{\exists\rho^*\ \textit{s.t.}\ \rho\to\rho^*\ \textit{and}\ [\theta]\rho^*=\rho'}$$

$$\begin{tikzcd}
\rho \arrow[r, dashed, "\to"] \arrow[d, "\hat{x}_1{:=}\tau_1"'] & \rho^* \arrow[d, dashed, "\hat{x}_1{:=}\tau_1"] \\
\bullet \arrow[r, dashed, "\to"] \arrow[d, dash, dotted] & \rho^*_2 \arrow[d, dash, dotted] \\
\bullet \arrow[r, dashed, "\to"] \arrow[d, "\hat{x}_n{:=}\tau_n"'] & \rho^*_n \arrow[d, dashed, "\hat{x}_n{:=}\tau_n"] \\
{[\theta]\rho} \arrow[r, "\to"] & \rho'
\end{tikzcd}$$

Figure 1: Proof Scheme

*Proof.* Induction on $\theta$, destruct $[\theta]\rho$, then lemma 7. See fig. 1 for the details: we obtain $\rho^*$ by consequently applying lemma 7 bottom-to-top to construct $\rho^*_n,\ldots,\rho^*_2,\rho^*$. The premises required for lemma 7 hold because $\vdash\theta$ and $\rho\,\mathsf{OK}$. □

**Corollary 4** (Context application preserves reduction)**.**

$$\frac{\vdash\theta\qquad\rho\to\rho'}{[\theta]\rho\to[\theta]\rho'}$$

*Proof.* Induction on $\theta$ using lemma 4. □

**Corollary 5** (Reduction preserves safety)**.**

$$\frac{\rho\,\mathsf{OK}\qquad\rho\to\rho'}{\rho'\,\mathsf{OK}}$$

*Proof.* Follows from lemma 9. □

**Definition 26** (Unification)**.** *The unification algorithm is defined as follows:*

**Base rules**

$$\frac{\hat{x}\in\theta\qquad\tau\,\mathsf{NF}\qquad\theta_{\hat{x}}\vdash\tau}{\theta\vdash\hat{x}\equiv\tau\dashv\theta\{\hat{x}\ :=\ (\hat{x}{:=}\tau)\}}\ \textsc{U-Add}\qquad\qquad\frac{(\hat{x}{:=}\tau)\in\theta}{\theta\vdash\hat{x}\equiv\tau\dashv\theta}\ \textsc{U-Keep}$$

**Reduction closure** `Ilya: could be packed using` ⇓

$$\frac{\rho\to\rho'\qquad\theta_1\vdash\rho'\equiv\tau\dashv\theta_2}{\theta_1\vdash\rho\equiv\tau\dashv\theta_2}\ \textsc{Red-L}\qquad\qquad\frac{\tau\to\tau'\qquad\theta_1\vdash\rho\equiv\tau'\dashv\theta_2\qquad\rho\,\mathsf{NF}}{\theta_1\vdash\rho\equiv\tau\dashv\theta_2}\ \textsc{Red-R}$$

**Congruence**

$$\frac{\theta_0 \vdash P_1 \equiv Q_1 \dashv \theta_1 \quad \ldots \quad \theta_{|F|-1} \vdash P_{|F|} \equiv Q_{|F|} \dashv \theta_{|F|} \qquad F\,P_1 \ldots P_{|F|}\ \mathsf{NF} \qquad F\,Q_1 \ldots Q_{|F|}\ \mathsf{NF}}{\theta_0 \vdash F\,P_1 \ldots P_{|F|} \equiv F\,Q_1 \ldots Q_{|F|} \dashv \theta_{|F|}}$$

$$\frac{\theta_1,\, x \vdash \vec{x}^{\,n}\rho \equiv \vec{x}^{\,n}\tau \dashv \theta_2}{\theta_1 \vdash x.\vec{x}^{\,n}\rho \equiv x.\vec{x}^{\,n}\tau \dashv \theta_2} \qquad\qquad \frac{\theta_1 \vdash \rho \equiv \tau \dashv \theta_2}{\theta_1 \vdash .\rho \equiv .\tau \dashv \theta_2}$$

We prove the soundness and completeness of the unification w.r.t. the equality defined above. Intuitively, soundness means that the output context produced by the unification algorithm does not make the terms non-unifiable.

**Lemma 13** (Unification soundness)**.**

$$\frac{\vdash \theta_1 \qquad \theta_1 \vdash \rho \equiv \tau \dashv \theta_2}{\theta_1 \longrightarrow \theta_2 \qquad \vdash \theta_2 \qquad \mathsf{gvars}(\theta_1) \vdash [\theta_2]\rho \equiv \tau}$$

*Proof.* `Ilya: TODO: update the proof`

□

**Lemma 14** (Unification completeness)**.**

$$\frac{\vdash \theta \qquad \rho\,\mathsf{OK} \qquad \mathsf{gvars}(\theta) \vdash [\theta]\rho \equiv \tau}{\forall \varphi \longrightarrow \theta.\ \exists \psi \longrightarrow \theta.\ \varphi \vdash \rho \equiv \tau \dashv \psi}$$

*Proof.* `Ilya: TODO: update the proof`

□

# 9 Typing

## 9.1 Declarative Type System

**Definition 27** (Declarative Typing Context)**.**

$$\Gamma \quad ::= \quad \cdot \ \mid\ \Gamma,\, x : A \ \mid\ \Gamma,\, \hat{x} : A$$

In the declarative system, we treat the existential variables $\hat{x}$ as regular variables. However, in the declarative system they cannot be added into the context, and must be there initially.

To make the typing decidable, we restrict the system in several ways. In particular, when we form $\forall x : A.\, X$, we require $x$ to belong to $\mathsf{FV}(X)$ and occur safely in $X$.

**Var and Annotation**

$$\frac{x : A \in \Gamma}{\Gamma \vdash_v x \Rightarrow A}\ \textsc{Var} \qquad \frac{\hat{x} : A \in \Gamma}{\Gamma \vdash_v \hat{x} \Rightarrow A}\ \textsc{VarE} \qquad \frac{}{\Gamma \vdash a\ \mathsf{vtype}}\ \textsc{VarT}$$

$$\frac{\Gamma \vdash_v v \Leftarrow A}{\Gamma \vdash_v v :: A \Rightarrow A}\ \textsc{Annot}$$

**Declarative Subsumption**

$$\frac{\Gamma \vdash_c t \Rightarrow X \qquad \Gamma \vdash X \leqslant^c Y}{\Gamma \vdash_c t \Leftarrow Y}\ \leqslant^c \qquad\qquad \frac{\Gamma \vdash_v v \Rightarrow A \qquad \Gamma \vdash A \leqslant^v B}{\Gamma \vdash_v v \Leftarrow B}\ \leqslant^v$$

### Type Formation

$$\frac{\Gamma \vdash A\,\mathsf{vtype}}{\Gamma \vdash {\uparrow} A\,\mathsf{ctype}}\ \mathcal{F} \qquad \frac{\Gamma \vdash X\ \mathsf{ctype}}{\Gamma \vdash {\downarrow} X\ \mathsf{vtype}}\ \mathcal{U} \qquad \frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma, x : A \vdash X\ \mathsf{ctype}}{\Gamma \vdash \Pi x : A.\, X\ \mathsf{ctype}}\ \Pi$$

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma, x : A \vdash X\ \mathsf{ctype} \qquad x \in^! X\ \mathsf{OK}}{\Gamma \vdash \forall x : A.\, X\ \mathsf{ctype}}\ \forall$$

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma \vdash_v v \Leftarrow A \qquad \Gamma \vdash_v w \Leftarrow A}{\Gamma \vdash \mathsf{eq} A\, v\, w\, \mathsf{vtype}}\ \mathsf{eq}$$

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma, x : A \vdash X\ \mathsf{ctype} \qquad \Gamma \vdash_c e \Leftarrow {\uparrow} A}{\Gamma \vdash (\mathsf{let}\ x : A\ := \ e\ \mathsf{in}\ X)\,\mathsf{ctype}}\ \textsc{Let-type}$$

### $\mathcal{F}$ and $\mathcal{U}$

$$\xcancel{\frac{\Gamma \vdash_c t \Rightarrow X \qquad \Gamma \vdash X \leqslant^c Y \qquad \Gamma \vdash Y \leqslant^c X}{\Gamma \vdash_v \{t\} \Leftarrow {\downarrow} Y}\ \mathcal{U}\mathrm{I}{\Leftarrow}} \qquad \frac{\Gamma \vdash_c t \Rightarrow X}{\Gamma \vdash_v \{t\} \Rightarrow {\downarrow} X}\ \mathcal{U}\mathrm{I}{\Rightarrow}$$

$$\xcancel{\frac{\Gamma \vdash_v v \Leftarrow {\downarrow} X}{\Gamma \vdash_c \mathsf{force}\, v \Leftarrow X}\ \mathcal{U}\mathrm{E}{\Leftarrow}} \qquad \frac{\Gamma \vdash_v v \Rightarrow {\downarrow} X}{\Gamma \vdash_c \mathsf{force}\, v \Rightarrow X}\ \mathcal{F}\mathrm{E}{\Rightarrow} \qquad \frac{\Gamma \vdash_v v \Leftarrow A}{\Gamma \vdash_c \mathsf{return}\, v \Leftarrow {\uparrow} A}\ \mathcal{F}\mathrm{I}{\Leftarrow}$$

$$\frac{\Gamma \vdash_v v \Rightarrow A}{\Gamma \vdash_c \mathsf{return}\, v \Rightarrow {\uparrow} A}\ \mathcal{F}\mathrm{I}{\Rightarrow}$$

### Let and Dependent Let

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma, x : A \vdash_c u \Rightarrow X \qquad \Gamma \vdash_c t \Leftarrow {\uparrow} A \qquad \Gamma \vdash X\ \mathsf{ctype}}{\Gamma \vdash_c \mathsf{let}\ x : A\ := \ t\ \mathsf{in}\ u \Rightarrow X}\ \textsc{Let}{\Rightarrow}$$

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma \vdash_c t \Leftarrow {\uparrow} A \qquad \Gamma \vdash X\ \mathsf{ctype} \qquad \Gamma, x : A \vdash_c u \Leftarrow X}{\Gamma \vdash_c \mathsf{let}\ x : A\ := \ t\ \mathsf{in}\ u \Leftarrow X}\ \textsc{Let}{\Leftarrow}$$

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma, x : A \vdash_c u \Rightarrow X \qquad \Gamma \vdash_c t \Leftarrow {\uparrow} A \qquad \Gamma, x : A \vdash X\ \mathsf{ctype}}{\Gamma \vdash_c \mathsf{dlet}\ x : A\ := \ t\ \mathsf{in}\ u \Rightarrow (\mathsf{let}\ x : A\ := \ t\ \mathsf{in}\ X)}\ \textsc{DLet}{\Rightarrow}$$

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma \vdash_c t \Leftarrow {\uparrow} A \qquad \Gamma, x : A \vdash X\ \mathsf{ctype} \qquad \Gamma, x : A \vdash_c u \Leftarrow X}{\Gamma \vdash_c \mathsf{dlet}\ x : A\ := \ t\ \mathsf{in}\ u \Leftarrow (\mathsf{let}\ x : A\ := \ t\ \mathsf{in}\ X)}\ \textsc{DLet}{\Leftarrow}$$

### $\forall$ and $\Pi$

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma, x : A \vdash_c t \Rightarrow X \qquad \Gamma, x : A \vdash X\ \mathsf{ctype}}{\Gamma \vdash_c \lambda x : A.\, t \Rightarrow \Pi x : A.\, X}\ \Pi\mathrm{I}{\Rightarrow}$$

$$\frac{\Gamma \vdash A\,\mathsf{vtype} \qquad \Gamma, x : A \vdash_c t \Rightarrow X \qquad \Gamma, x : A \vdash X\ \mathsf{ctype} \qquad x \in^! X\ \mathsf{OK}}{\Gamma \vdash_c \hat{\lambda} x : A.\, t \Rightarrow \forall x : A.\, X}\ \forall\mathrm{I}{\Rightarrow}$$

**Equality**

$$\frac{\Gamma \vdash_v v \Rightarrow A}{\Gamma \vdash_v \mathsf{refl}_v \Rightarrow \mathsf{eq}\,A\,v\,v}\ \mathsf{eqI}$$

$$\frac{\begin{array}{c}\Gamma \vdash_v v \Rightarrow \mathsf{eq}\,A\,w_1\,w_2 \\ \Gamma, x : A, p : \mathsf{eq}\,A\,w_1\,x \vdash X\ \mathsf{ctype} \qquad \Gamma \vdash_c t \Leftarrow X\{x := w_1\}\{p := \mathsf{refl}_{w_1}\}\end{array}}{\Gamma \vdash_c \mathsf{rec}_{\mathsf{eq}}^{x.p.X}(v, t) \Rightarrow X\{x := w_2\}\{p := v\}}\ \mathsf{eqE}{\Leftarrow}$$

**Declarative Polymorphic Application**

$$\frac{\Gamma \vdash_c t \Rightarrow Y \qquad \Gamma \vdash Y \bullet v \Rrightarrow X}{\Gamma \vdash_c t\,v \Rightarrow X}\ \Pi\mathrm{E} \qquad\qquad \frac{\Gamma \vdash_v w \Leftarrow A \qquad \Gamma \vdash X\{x := w\} \bullet v \Rrightarrow Y}{\Gamma \vdash \forall x : A.\, X \bullet v \Rrightarrow Y}\ \forall\textsc{App}$$

$$\frac{\xcancel{\Gamma, x : A} \vdash X \bullet v \Rrightarrow Y \qquad x \in^{!} Y\ \mathsf{OK}}{\Gamma \vdash \forall x : A.\, X \bullet v \Rrightarrow \forall x : A.\, Y}\ \forall\textsc{App-Regen}$$

$$\frac{\Gamma \vdash_v v \Leftarrow A \qquad \texttt{Ilya:  X is already WF}}{\Gamma \vdash \Pi x : A.\, X \bullet v \Rrightarrow X\{x := v\}}\ \Pi\textsc{App}$$

**Declarative Subtyping**

$$\frac{\Gamma \vdash A_2 \leqslant^v A_1 \qquad \Gamma, x : A_2 \vdash X_1 \leqslant^c X_2}{\Gamma \vdash \Pi x : A_1.\, X_1 \leqslant^c \Pi x : A_2.\, X_2}\ \leqslant \Pi \qquad\qquad \frac{}{\Gamma \vdash a \leqslant^v a}\ \leqslant\textsc{Var}$$

$$\frac{\Gamma \vdash X_1 \leqslant^c X_2 \qquad \Gamma \vdash X_2 \leqslant^c X_1}{\Gamma \vdash {\downarrow} X_1 \leqslant^v {\downarrow} X_2}\ \leqslant \mathcal{U} \qquad\qquad \frac{\Gamma \vdash A_1 \leqslant^v A_2 \qquad \Gamma \vdash A_2 \leqslant^v A_1}{\Gamma \vdash {\uparrow} A_1 \leqslant^c {\uparrow} A_2}\ \leqslant \mathcal{F}$$

$$\frac{\Gamma \vdash A \leqslant^v B \qquad \mathsf{unt}(\Gamma) \vdash v_1 \equiv v_2 \qquad \mathsf{unt}(\Gamma) \vdash w_1 \equiv w_2}{\Gamma \vdash \mathsf{eq}\,A\,v_1\,w_1 \leqslant^v \mathsf{eq}\,B\,v_2\,w_2}\ \leqslant\textsc{Eq}$$

$$\frac{\Gamma \vdash_v v \Leftarrow A \qquad \Gamma \vdash X\{x := v\} \leqslant^c Y}{\Gamma \vdash (\forall x : A.\, X) \leqslant^c Y}\ \forall \leqslant \qquad\qquad \frac{\Gamma, y : A \vdash X \leqslant^c Y}{\Gamma \vdash X \leqslant^c (\forall y : A.\, Y)}\ \leqslant \forall$$

$$\frac{e \Downarrow \mathsf{return}\,v \qquad \Gamma \vdash X\{x := v\} \leqslant^c Y}{\Gamma \vdash (\mathsf{let}\ x : A := e\ \mathsf{in}\ X) \leqslant^c Y}\ \textsc{let}{\leqslant} \qquad\qquad \frac{e \Downarrow \mathsf{return}\,v \qquad \Gamma \vdash X \leqslant^c Y\{y := v\}}{\Gamma \vdash X \leqslant^c (\mathsf{let}\ y : A := e\ \mathsf{in}\ Y)}\ {\leqslant}\textsc{let}$$

$$\frac{\Gamma \vdash A \leqslant^v B \qquad \Gamma \vdash B \leqslant^v A \qquad \mathsf{unt}(\Gamma) \vdash e_1 \equiv e_2 \qquad \Gamma, x : A \vdash X \leqslant^c Y}{\Gamma \vdash \mathsf{let}\ x : A := e_1\ \mathsf{in}\ X \leqslant^c \mathsf{let}\ x : B := e_2\ \mathsf{in}\ Y}\ \textsc{let}{\leqslant}\textsc{let}$$

# 10 Algorithmic Type System

In the declarative system, there are two places where the oracle is involved (highlighted red): ($\forall \leqslant$) and ($\forall$App). To make the system algorithmic, we must amend these rules.

**Definition 28** (Algorithmic Typing Context)**.**

$$\Theta \quad ::= \quad \cdot \ \mid\ \Theta,\, x : A \ \mid\ \Theta,\, \hat{x} : A \ \mid\ \Theta,\, (\hat{x} {:=} v)$$

**Definition 29** (Context extension)**.**

$$\frac{\Theta_1 \longrightarrow \Theta_2 \qquad e \in \{x : A, \hat{x} : A, (\hat{x}{:=}v)\}}{\Theta_1,\, e \longrightarrow \Theta_2,\, e} \qquad\qquad \frac{\Theta_1 \longrightarrow \Theta_2 \qquad \mathsf{GR}\, v \qquad |\Theta_2| \vdash_v v \Leftarrow A}{\Theta_1,\, \hat{x} : A \longrightarrow \Theta_2,\, (\hat{x}{:=}v)}$$

**Definition 30** (Context Truncation)**.** *Truncation maps an algorithmic context into a declarative one by removing the resolved existential variables. We will keep the invariant that the resolved variables cannot occur in the types of the unresolved ones, as such, truncation preserves well-formedness.*

$$|\cdot| = \cdot \qquad\qquad |\Theta,\, \hat{x} : A| = |\Theta|,\, \hat{x} : A$$

$$|\Theta,\, x : A| = |\Theta|,\, x : A \qquad\qquad |\Theta,\, (\hat{x}{:=}v)| = |\Theta|$$

**Algorithmic Polymorphic Application**

$$\frac{\Theta_1, \hat{x} : A \vDash X\{x := \hat{x}\} \bullet v \Rrightarrow Y \Dashv \Theta_2, (\hat{x}{:=}w) \qquad \Theta_2 \vDash_v w \Leftarrow A \Dashv \Theta_3}{\Theta_1 \vDash \forall x : A.\, X \bullet v \Rrightarrow [\Theta_3]Y \Dashv \Theta_3}\ \text{A}\forall\textsc{App}$$

$$\frac{\Theta_1, \hat{x} : A \vDash X\{x := \hat{x}\} \bullet v \Rrightarrow Y \Dashv \Theta_2, \hat{x} : A' \qquad \hat{x} \in^! Y\, \mathsf{OK}}{\Theta_1 \vDash \forall x : A.\, X \bullet v \Rrightarrow \forall x : A'.\, Y\{\hat{x} := x\} \Dashv \Theta_2}\ \text{A}\forall\textsc{App-Regen}$$

$$\frac{\Theta_1 \vDash_v v \Leftarrow A \Dashv \Theta_2}{\Theta_1 \vDash \Pi x : A.\, X \bullet v \Rrightarrow [\Theta_2]X\, \{x := v\} \Dashv \Theta_2}\ \text{A}\Pi\textsc{App}$$

**Algorithmic Value Type Checking**

$$\frac{\Theta_1 \vDash_v v \Rightarrow A \qquad \Theta_1 \vDash A \leqslant^v B \Dashv \Theta_2}{\Theta_1 \vDash_v v \Leftarrow B \Dashv \Theta_2}\ \text{A}{\leqslant^v}$$

$$\xcancel{\frac{\Theta_1 \vDash_c t \Rightarrow X \qquad \Theta_1 \vDash Y \leqslant^c X \Dashv \Theta_2 \qquad \Theta_2 \vDash X \leqslant^c [\Theta_2]Y \Dashv \Theta_3}{\Theta_1 \vDash_v \{t\} \Leftarrow {\downarrow}Y \Dashv \Theta_3}\ \mathcal{U}\text{I}{\Leftarrow}}$$

**Algorithmic Type Formation, Type Inference, Computational Type Checking** The rule VarE is excluded from the algorithmic system. This is because the existential variables $\hat{x}$ are to be initialized with some ground terms.

These rules are almost identical to their declarative counterparts. The difference is that the context for the algorithmic judgments is algorithmic ($\Theta$, not $\Gamma$); and whenever Value Type Checking or Polymorphic Application judgment is invoked in the premise, we ignore the output algorithmic context.

**Algorithmic Subtyping**

$$\frac{\Theta_1 \vDash A_2 \leqslant^v A_1 \Dashv \Theta_2 \qquad \Theta_2, x : A_2 \vDash X_1 \leqslant^c X_2 \Dashv \Theta_3, x : A_2}{\Theta_1 \vDash \Pi x : A_1.\, X_1 \leqslant^c \Pi x : A_2.\, X_2 \Dashv \Theta_3} \leqslant \Pi$$

$$\frac{\Theta_1 \vDash X_2 \leqslant^c X_1 \Dashv \Theta_2 \qquad \Theta_2 \vDash X_1 \leqslant^c [\Theta_2] X_2 \Dashv \Theta_3}{\Theta_1 \vDash {\downarrow} X_1 \leqslant^v {\downarrow} X_2 \Dashv \Theta_3} \leqslant \mathcal{U} \quad \frac{\Theta_1 \vDash A_2 \leqslant^v A_1 \Dashv \Theta_2 \qquad \Theta_2 \vDash [\Theta_2] A_1 \leqslant^v A_2 \Dashv \Theta_3}{\Theta_1 \vDash {\uparrow} A_1 \leqslant^c {\uparrow} A_2 \Dashv \Theta_3} \leqslant \mathcal{F}$$

$$\frac{\Theta_1 \vDash A \leqslant^v B \Dashv \Theta_2 \qquad \mathsf{unt}(\Theta_2) \vdash v_2 \equiv v_1 \dashv \theta_1 \qquad \theta_1 \vdash w_2 \equiv w_1 \dashv \theta_2}{\Theta_1 \vDash \mathsf{eq} A\, v_1\, w_1 \leqslant^v \mathsf{eq} B\, v_2\, w_2 \Dashv [\theta_2]\Theta_2} \leqslant\text{EQ}$$

$$\frac{\Theta_1, \hat{x} : A \vDash X\{x \rightsquigarrow \hat{x}\} \leqslant^c Y \Dashv \Theta_2, (\hat{x}{:=}v) \qquad \Theta_2 \vDash_v v \Leftarrow [\Theta_2]A \Dashv \Theta_3}{\Theta_1 \vDash \forall x : A.\, X \leqslant^c Y \Dashv \Theta_3} \forall\leqslant \quad \frac{\Theta_1, y : A \vDash X \leqslant^c Y \Dashv \Theta_2}{\Theta_1 \vDash X \leqslant^c (\forall y : A.\, Y) \Dashv \Theta_2} \leqslant \forall$$

$$\frac{e \Downarrow \mathsf{return}\; v \qquad \Theta_1 \vDash X\{x := v\} \leqslant^c Y \Dashv \Theta_2}{\Theta_1 \vDash \mathsf{let}\; x : A := e\; \mathsf{in}\; X \leqslant^c Y \Dashv \Theta_2} \text{LET}\leqslant \quad \frac{e \Downarrow \mathsf{return}\; v \qquad \Theta_1 \vDash X \leqslant^c Y\{y := v\} \Dashv \Theta_2}{\Theta_1 \vDash X \leqslant^c (\mathsf{let}\; y : A := e\; \mathsf{in}\; Y) \Dashv \Theta_2} \leqslant\text{LET}$$

$$\frac{\begin{array}{c}\Theta_1 \vDash B \leqslant^v A \Dashv \Theta_2 \qquad \Theta_2 \vDash [\Theta_2] A \leqslant^v B \Dashv \Theta_3 \\ \mathsf{unt}(\Theta_3) \vdash e_1 \equiv e_2 \dashv \theta \qquad [\theta]\Theta_3, x : [\Theta_2]A \vDash X \leqslant^c Y \Dashv \Theta_4, x : \cdot\end{array}}{\Theta_1 \vDash \mathsf{let}\; x : A := e_1\; \mathsf{in}\; X \leqslant^c \mathsf{let}\; x : B := e_2\; \mathsf{in}\; Y \Dashv \Theta_4} \text{LET}\leqslant\text{LET}$$

# 11 The substitution lemma

**Lemma 15** (The substitution lemma)**.**

$$\frac{\Gamma, x : A \vdash X\; \mathsf{ctype} \qquad \Gamma \vdash_v v \Leftarrow A}{\Gamma \vdash X\{x := v\}\; \mathsf{ctype}}$$

**Corollary 6** (The safety of context extension)**.**

$$\frac{|\Theta_1| \vdash A\; \mathsf{vtype} \qquad \Theta_1 \longrightarrow \Theta_2}{|\Theta_1| \vdash [\Theta_2]A\; \mathsf{vtype}} \qquad \frac{|\Theta_1| \vdash_v v \Leftarrow A \qquad \Theta_1 \longrightarrow \Theta_2}{|\Theta_1| \vdash_v v \Leftarrow [\Theta_2]A} \qquad \frac{|\Theta_1| \vdash_c t \Leftarrow X \qquad \Theta_1 \longrightarrow \Theta_2}{|\Theta_1| \vdash_c t \Leftarrow [\Theta_2]X}$$

# 12 Soundness

**Lemma 16** (Algorithmic Type System Soundness)**.**

$$\dfrac{\vdash \Theta_1 \qquad |\Theta_1| \vdash A\ \mathsf{vtype} \qquad |\Theta_1| \vdash B\ \mathsf{vtype} \qquad \mathsf{GR}\ A \qquad \Theta_1 \vDash A \leqslant^v B \Dashv \Theta_2}{\vdash \Theta_2 \qquad \Theta_1 \longrightarrow \Theta_2 \qquad \mathsf{GR}\ [\Theta_2]B \qquad |\Theta_1| \vdash A \leqslant^v [\Theta_2]B}$$

$$\dfrac{\vdash \Theta_1 \qquad |\Theta_1| \vdash X\ \mathsf{ctype} \qquad |\Theta_1| \vdash Y\ \mathsf{ctype} \qquad \mathsf{GR}\ Y \qquad \Theta_1 \vDash X \leqslant^c Y \Dashv \Theta_2}{\vdash \Theta_2 \qquad \Theta_1 \longrightarrow \Theta_2 \qquad \mathsf{GR}\ [\Theta_2]X \qquad |\Theta_1| \vdash [\Theta_2]X \leqslant^c Y}$$

$$\dfrac{\vdash \Theta_1 \qquad |\Theta_1| \vdash X\ \mathsf{ctype} \qquad \mathsf{GR}\ v \qquad \Theta_1 \vDash X \bullet v \Rrightarrow Y \Dashv \Theta_2}{\begin{array}{c}\vdash \Theta_2 \qquad \Theta_1 \longrightarrow \Theta_2 \qquad |\Theta_1| \vdash Y\ \mathsf{ctype} \qquad [\Theta_2]Y = Y \qquad \mathsf{EV}(Y) \subseteq \mathsf{EV}(X) \\ \forall \Omega\ \ s.t.\ \ \Theta_2 \longrightarrow \Omega,\ |\Theta_1| \vdash [\Omega]X \bullet v \Rrightarrow [\Omega]Y\end{array}}$$

$$\dfrac{\vdash \Theta_1 \qquad |\Theta_1| \vdash A\ \mathsf{vtype} \qquad \mathsf{GR}\ v \qquad \Theta_1 \vDash_v v \Leftarrow A \Dashv \Theta_2}{\vdash \Theta_2 \qquad \Theta_1 \longrightarrow \Theta_2 \qquad \mathsf{GR}\ [\Theta_2]A \qquad |\Theta_1| \vdash_v v \Leftarrow [\Theta_2]A}$$

$$\dfrac{\vdash \Theta \qquad |\Theta| \vdash X\ \mathsf{ctype} \qquad \mathsf{GR}\ t \qquad \mathsf{GR}\ X \qquad \Theta \vDash_c t \Leftarrow X}{|\Theta| \vdash_c t \Leftarrow X}$$

$$\dfrac{\vdash \Theta \qquad \mathsf{GR}\ v \qquad \Theta \vDash_v v \Rightarrow A}{\mathsf{GR}\ A \qquad |\Theta| \vdash A\ \mathsf{vtype} \qquad |\Theta| \vdash_v v \Rightarrow A} \qquad \dfrac{\vdash \Theta \qquad \mathsf{GR}\ t \qquad \Theta \vDash_c t \Rightarrow X}{\mathsf{GR}\ X \qquad |\Theta| \vdash X\ \mathsf{ctype} \qquad |\Theta| \vdash_c t \Rightarrow X}$$

$$\dfrac{\vdash \Theta \qquad \mathsf{GR}\ A \qquad \Theta \vDash A\ \mathsf{vtype}}{|\Theta| \vdash A\ \mathsf{vtype}} \qquad \dfrac{\vdash \Theta \qquad \mathsf{GR}\ X \qquad \Theta \vDash X\ \mathsf{ctype}}{|\Theta| \vdash X\ \mathsf{ctype}}$$

*Proof.*

**Polymorphic Application** Induction on $\Theta_1 \vDash X \bullet v \Rrightarrow Y \Dashv \Theta_2$.

A∀App Considering this case implies unification $\Theta_1 \vDash X \bullet v \Rrightarrow Y \Dashv \Theta_2$ with $\Theta_1 \vDash \forall x : A.\, X \bullet v \Rrightarrow [\Theta_3]Y \Dashv \Theta_3$ which entails the following specializations:

| original term | specialized as |
|---|---|
| $X$ | $\forall x : A.\, X$ |
| $\Theta_2$ | $\Theta_3$ |
| $Y$ | $[\Theta_3]Y$ |

Given:

1. $\vdash \Theta_1$
2. $|\Theta_1| \vdash \forall x : A.\, X$ ctype
3. GR $v$
4. $\Theta_3 \longrightarrow \Omega$
5. $\Theta_1 \vDash \forall x : A.\, X \bullet v \Rrightarrow Y \Dashv \Theta_3$ By inversion,
   *a.* $\Theta_1, \hat{x} : A \vDash X\{x := \hat{x}\} \bullet v \Rrightarrow Y \Dashv \Theta_2, (\hat{x}{:=}w)$ Applying the IH, `Ilya: some extra lemmas required to prove the premises`
      *i.* $\vdash \Theta_2, (\hat{x}{:=}w)$
      *ii.* $\Theta_1, \hat{x} : A \longrightarrow \Theta_2, (\hat{x}{:=}w)$
         1. $\Theta_1 \longrightarrow \Theta_2$
         2. $|\Theta_2| \vdash_v w \Leftarrow A$

*iii.* $|\Theta_1, \hat{x} : A| \vdash Y\ \mathsf{ctype}$

*iv.* $[\Theta_2, (\hat{x}{:=}w)]Y = Y$. It means $\hat{x} \notin Y$, and thus, $[\Omega, (\hat{x}{:=}w)]Y = [\Omega]Y$.

*v.* $\mathsf{EV}(Y) \subseteq \mathsf{EV}(X\{x := \hat{x}\})$, and thus, because $\hat{x} \notin Y$, we have $\mathsf{EV}(Y) \subseteq \mathsf{EV}(X)$.

*vi.* specializing "$\forall\Omega$" with $\Omega, (\hat{x}{:=}w)$, we have $|\Theta_1, \hat{x} : A| \vdash [\Omega, (\hat{x}{:=}w)]X\{x := \hat{x}\} \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} [\Omega, (\hat{x}{:=}w)]Y$ and since the terms do not contain $\hat{x}$, $|\Theta_1| \vdash [\Omega]X\{x := w\} \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} [\Omega]Y$

*b.* $\Theta_2 \vDash_v w \Leftarrow A \Dashv \Theta_3$. Applying the IH, we also have:

*i.* $\vdash \Theta_3$

*ii.* $\Theta_2 \longrightarrow \Theta_3$

*iii.* $\mathsf{GR}\,[\Theta_3]A$, and since $\Theta_3 \longrightarrow \Omega$, $[\Theta_3]A = [\Omega]A$

*iv.* $|\Theta_2| \vdash_v w \Leftarrow [\Theta_3]A$, and since $|\Theta_1| \supseteq |\Theta_2|$, $|\Theta_1| \vdash_v w \Leftarrow [\Theta_3]A$

We prove:

1. $\vdash \Theta_3$ by point $(5.b.i)$
2. $\Theta_1 \longrightarrow \Theta_3$ from point $(5.a.ii.1)$ and point $(5.b.ii)$ by transitivity
3. $|\Theta_1| \vdash [\Theta_3]Y\ \mathsf{ctype}$. From point $(5.a.iii)$ and because $\hat{x} \notin Y$, we have $|\Theta_1| \vdash Y\ \mathsf{ctype}$. Then because $\Theta_1 \longrightarrow \Theta_3$ and corollary 6, we have the wanted property.
4. $[\Theta_3][\Theta_3]Y = [\Theta_3]Y$ by idempotency of context substitution.
5. $[\Theta_3]Y \subseteq \mathsf{EV}(\forall x : A.\, X)$ by point $(5.a.v)$.
6. $|\Theta_1| \vdash [\Omega]\forall x : A.\, X \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} [\Omega][\Theta_3]Y$. Since $\Theta_3 \longrightarrow \Omega$, and by substitution congruence, we rewrite the statement: $|\Theta_1| \vdash \forall x : [\Omega]A.\, [\Omega]X \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} [\Omega]Y$ and apply the declarative $\forall App$.

   *a.* $|\Theta_1| \vdash_v w \Leftarrow [\Omega]A$. From point $(5.b.iv)$ and corollary 6, since $\Theta_3 \longrightarrow \Omega$

   *b.* $|\Theta_1| \vdash [\Omega]X\{x := w\} \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} [\Omega]Y$ by point $(5.a.vi)$.

A$\forall$App-Regen Considering this case implies unification $\Theta_1 \vDash X \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} Y \Dashv \Theta_2$ with $\Theta_1 \vDash \forall x : A.\, X \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} \forall x : A'.\, Y\{\hat{x} := x\} \Dashv \Theta_2$ which entails the following specializations:

| original term | specialized as |
|---|---|
| $X$ | $\forall x : A.\, X$ |
| $Y$ | $\forall x : A'.\, Y\{\hat{x} := x\}$ |

Given:

1. $\vdash \Theta_1$
2. $|\Theta_1| \vdash \forall x : A.\, X\ \mathsf{ctype}$. Then by inversion,

   *a.* $|\Theta_1| \vdash A\ \mathsf{vtype}$

   *b.* $|\Theta_1|, x : A \vdash X\ \mathsf{ctype}$

   *c.* $x \in^! X\ \mathsf{OK}$
3. $\mathsf{GR}\, v$
4. $\Theta_2 \longrightarrow \Omega$
5. $\Theta_1 \vDash \forall x : A.\, X \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} \forall x : A'.\, Y\{\hat{x} := x\} \Dashv \Theta_2$ By inversion,

   *a.* $\hat{x} \in^! Y\ \mathsf{OK}$

   *b.* $\Theta_1, \hat{x} : A \vDash X\{x := \hat{x}\} \bullet v \mathrel{\Rightarrow\mkern-14mu\Rightarrow} Y \Dashv \Theta_2, \hat{x} : A'$ Then we apply the induction hypothesis Ilya: TODO: Prove the IH premises. We will probably need to add something like $[\Theta_1]X = X$ to the premises to ensure the well-formedness of $\Theta_1, \hat{x} : A$

   *i.* $\vdash \Theta_2, \hat{x} : A'$

   *ii.* $\Theta_1, \hat{x} : A \longrightarrow \Theta_2, \hat{x} : A'$ , which implies that

   1. $\Theta_1 \longrightarrow \Theta_2$
   2. $A' = [\Theta_2]A$

   *iii.* $|\Theta_1, \hat{x} : A| \vdash Y\ \mathsf{ctype}$

   *iv.* $[\Theta_2, \hat{x} : A']Y = Y$. Hence, $[\Theta_2]Y = Y$

*v.* $\mathsf{EV}(Y) \subseteq \mathsf{EV}(X\{x := \hat{x}\})$

*vi.* $\forall \Omega$ s.t. $\Theta_2, \hat{x} : A' \longrightarrow \Omega$, $|\Theta_1, \hat{x} : A| \vdash [\Omega]X\{x := \hat{x}\} \bullet v \Rrightarrow [\Omega]Y$. We specialize it with $\Omega := \Omega, \hat{x} : [\Omega]A'$ and obtain: $|\Theta_1, \hat{x} : A| \vdash [\Omega, \hat{x} : [\Omega]A']X\{x := \hat{x}\} \bullet v \Rrightarrow [\Omega, \hat{x} : [\Omega]A']Y$ Unfolding it further, we have: $|\Theta_1|, \hat{x} : A \vdash [\Omega]X\{x := \hat{x}\} \bullet v \Rrightarrow [\Omega]Y$. Then we rename $\hat{x}$ into $x$ (recall that we treat the existential variables as the normal ones in the declarative system): $|\Theta_1|, x : A \vdash [\Omega]X \bullet v \Rrightarrow [\Omega]Y\{\hat{x} := x\}$, then by the corollary from the substitution lemma, since $\Theta_1 \longrightarrow \Omega$, we have: $|\Theta_1|, x : [\Omega]A \vdash [\Omega]X \bullet v \Rrightarrow [\Omega]Y\{\hat{x} := x\}$.

We prove:

1. $\vdash \Theta_2$ from point $(5.b.i)$
2. $\Theta_1 \longrightarrow \Theta_2$ by point $(5.b.ii.1)$
3. $|\Theta_1| \vdash \forall x : A'.\, Y\{\hat{x} := x\}\, \mathsf{ctype}$. We apply the declarative rule $\forall$:
   - *a.* $\Theta_1 \vdash A'\, \mathsf{vtype}$. From point $(5.b.ii.2)$, it suffices to prove $\Theta_1 \vdash [\Theta_2]A\, \mathsf{vtype}$, which follows from corollary 6 because $\Theta_1 \longrightarrow \Theta_2$.
   - *b.* $|\Theta_1|, x : A' \vdash Y\{\hat{x} := x\}\, \mathsf{ctype}$ from point $(5.b.iii)$ and the substitution lemma
   - *c.* $x \in^! Y\{\hat{x} := x\}\, \mathsf{OK}$ by point $(5.a)$
4. $[\Theta_2]\forall x : A'.\, Y\{\hat{x} := x\} = \forall x : A'.\, Y\{\hat{x} := x\}$. Because $[\Theta_2]A' = [\Theta_2][\Theta_2]A = A'$ and point $(5.b.iv)$
5. $\mathsf{EV}(\forall x : A'.\, Y\{\hat{x} := x\}) \subseteq \mathsf{EV}(\forall x : A.\, X)$. Because $\mathsf{EV}(\forall x : A'.\, Y\{\hat{x} := x\}) \subseteq \mathsf{EV}(A') \cup \mathsf{EV}(Y)\backslash\{\hat{x}\} \subseteq \mathsf{EV}(A') \cup \mathsf{EV}(X\{x := \hat{x}\})\backslash\{\hat{x}\} \equiv \mathsf{EV}([\Theta_2]A) \cup \mathsf{EV}(X)\backslash\{x\} \subseteq \mathsf{EV}(A) \cup \mathsf{EV}(X)\backslash\{x\}$.
6. $|\Theta_1| \vdash [\Omega]\forall x : A.\, X \bullet v \Rrightarrow [\Omega]\forall x : A'.\, Y\{\hat{x} := x\}$ By context application congruence, it is equal to $|\Theta_1| \vdash \forall x : [\Omega]A.\, [\Omega]X \bullet v \Rrightarrow \forall x : [\Omega]A'.\, [\Omega]Y\{\hat{x} := x\}$ We apply the declarative rule $\forall$App-Regen. Thus, it suffices to prove:
   - *a.* $|\Theta_1|, x : [\Omega]A \vdash [\Omega]X \bullet v \Rrightarrow [\Omega]Y\{\hat{x} := x\}$ by point $(5.b.vi)$;
   - *b.* $x \in^! [\Omega]Y\{\hat{x} := x\}\, \mathsf{OK}$ by point $(5.a)$ and lemma 6

AΠApp Considering this case implies unification $\Theta_1 \models X \bullet v \Rrightarrow Y \dashv \Theta_2$ with $\Theta_1 \models \Pi x : A.\, X \bullet v \Rrightarrow [\Theta_2]X\{x := v\} \dashv \Theta_2$

which entails the following specializations:

| original term | specialized as |
|---|---|
| $X$ | $\Pi x : A.\, X$ |
| $Y$ | $[\Theta_2]X\{x := v\}$ |

Given:

1. $\vdash \Theta_1$
2. $|\Theta_1| \vdash \Pi x : A.\, X\, \mathsf{ctype}$ Then by corollary 6, $|\Theta_1| \vdash [\Theta_2]\Pi x : A.\, X\, \mathsf{ctype}$. Hence, by inversion,
   - *a.* $|\Theta_1| \vdash [\Theta_2]A\, \mathsf{vtype}$
   - *b.* $|\Theta_1|, x : [\Theta_2]A \vdash [\Theta_2]X\, \mathsf{ctype}$
3. $|\Theta_1| \vdash \Pi x : A.\, X\, \mathsf{ctype}$
4. $\mathsf{GR}\, v$
5. $\Theta_1 \models \Pi x : A.\, X \bullet v \Rrightarrow [\Theta_2]X\{x := v\} \dashv \Theta_2$. Then by inversion, $\Theta_1 \models_v v \Leftarrow A \dashv \Theta_2$. Then we applying the induction hypothesis `Ilya: the IH premises are straightforward`
   - *a.* $\vdash \Theta_2$
   - *b.* $\Theta_1 \longrightarrow \Theta_2$
   - *c.* $\mathsf{GR}\, [\Theta_2]A$
   - *d.* $|\Theta_1| \vdash_v v \Leftarrow [\Theta_2]A$
6. $\Theta_2 \longrightarrow \Omega$

We prove:

1. $\vdash \Theta_2$ by point $(5.a)$
2. $\Theta_1 \longrightarrow \Theta_2$ by point $(5.b)$
3. $|\Theta_1| \vdash [\Theta_2]X\{x := v\}\,\mathsf{ctype}$. From lemma 15, it suffices to prove
   *a.* $|\Theta_1|, x : [\Theta_2]A \vdash [\Theta_2]X\,\mathsf{ctype}$ by point $(2.b)$
   *b.* $|\Theta_1| \vdash_v v \Leftarrow [\Theta_2]A$ by point $(5.d)$
4. $[\Theta_2][\Theta_2]X\{x := v\} = [\Theta_2]X\{x := v\}$ by idempotency of context application
5. $\mathsf{EV}([\Theta_2]X\{x := v\}) \subseteq \mathsf{EV}(\Pi x : A.\, X)$ because $\mathsf{EV}([\Theta_2]X\{x := v\}) \subseteq \mathsf{EV}(X)\backslash x \subseteq \mathsf{EV}(\Pi x : A.\, X)$
6. $|\Theta_1| \vdash [\Omega]\Pi x : A.\, X \bullet v \Rrightarrow [\Omega][\Theta_2]X\{x := v\}$ Since $v$ is ground, the application of $\Omega$ commutes with the substitution $v$ for $x$. Then since $\Theta_2 \longrightarrow \Omega$ and by congruence of the context application, it suffices to prove $|\Theta_1| \vdash \Pi x : [\Omega]A.\, [\Omega]X \bullet v \Rrightarrow [\Omega]X\{x := v\}$ Then applying the declarative rule $(\Pi\text{App})$, it suffices to prove $|\Theta_1| \vdash_v v \Leftarrow [\Omega]A$, which follows from point $(5.d)$ and corollary 6

**Value Subtyping** Induction on $\Theta_1 \models A \leqslant^v B \dashv \Theta_2$.

$\leqslant$Eq Considering this case implies unification $\Theta_1 \models A \leqslant^v B \dashv \Theta_2$ with $\Theta_1 \models \mathsf{eq} A\, v_1\, w_1 \leqslant^v \mathsf{eq} B\, v_2\, w_2 \dashv [\theta_2]\Theta_2$ which entails the following specializations:

| original term | specialized as |
|---|---|
| $A$ | $\mathsf{eq} A\, v_1\, w_1$ |
| $B$ | $\mathsf{eq} B\, v_2\, w_2$ |
| $\Theta_2$ | $[\theta]\Theta_2$ |

□

# 13 Completeness

# 14 The Safe Variable Invariant

In section 10, we will prove the exhaustiveness of the algorithmic subtyping. For this purpose, we need to provide a certain invariant of how the existential variables are distributed in the well-formed types.

**Definition 31** (The Trace of the Term)**.** *Suppose $\rho$ is a term. Let us define $\mathsf{tr}(\rho)$ as*

$$\mathsf{tr}(\rho) = \mathsf{EV}(\Downarrow \rho)$$

**Definition 32** (The Contextual Background of the Term)**.** *Suppose $\Gamma$ is a context, $\rho$ is a term. Let us define $\mathsf{bg}(\Gamma, \rho)$ as the minimal set of ground variables, such that*

1. $\mathsf{FV}(\Downarrow \rho)\backslash\mathsf{EV}(\Downarrow \rho) \subseteq \mathsf{bg}(\Gamma, \rho)$
2. *if $x \in \mathsf{bg}(\Gamma, \rho)$ and $(x : \pi) \in \Gamma$ then $\mathsf{FV}(\Downarrow \pi)\backslash\mathsf{EV}(\Downarrow \pi) \subseteq \mathsf{bg}(\Gamma, \rho)$.*

*Intuitively, we take an empty set, add all the non-existential variables of $\rho$ into it, and recursively repeat this process for the types of the added variables.*

**Definition 33** (The Contextual Trace of the Term)**.** *Suppose $\Gamma$ is a context, $\rho$ is a term. Let us define $\mathsf{tr}(\Gamma, \rho)$ as*

$$\mathsf{tr}(\Gamma, \rho) = \mathsf{tr}(\rho) \cup \bigcup_{(x:\pi) \leftarrow \Gamma,\, x \leftarrow \mathsf{bg}(\Gamma, \rho)} \mathsf{tr}(\pi)$$

*Ilya: the old definition:*

$$\mathsf{tr}(\Gamma, \rho) = \mathsf{tr}(\rho) \cup \{\mathsf{tr}(\pi) \mid (x : \pi) \leftarrow \Gamma,\, x \leftarrow \mathsf{FV}(\Downarrow \rho)\backslash\mathsf{EV}(\Downarrow \rho)\}$$

## 14.1 Trace Transferred from Types

**Definition 34** (Reticent Terms Syntax)**.**

$$\begin{array}{lll} \textit{Reticent Values} & r_v & ::= \quad \hat{x} \ \mid \ \{r_c\} \\ \textit{Reticent Computations} & r_c & ::= \quad \mathsf{return}\, r_v \ \mid \ \mathsf{let}\ x : A \ := \ t \ \mathsf{in}\ r_c \ \mid \ \mathsf{dlet}\ x : A \ := \ t \ \mathsf{in}\ r_c \end{array}$$

**Proposition 5** (Inferred Types of Reticent Terms)**.**

$$\frac{\Gamma \vdash_v r_v \Rightarrow A}{r_v = \hat{x} \quad \textit{or} \quad A = \downarrow X} \qquad\qquad \frac{\Gamma \vdash_c r_c \Rightarrow X}{X = \uparrow A \quad \textit{or} \quad X = \mathsf{let}\ x : A \ := \ t \ \mathsf{in}\ Y}$$

**Lemma 17** (Inferred Types Trace)**.** *Intuitively, the non-reticent terms (contextually) expose the trace of the inferred types.*

$$\frac{v\,\mathsf{OK} \qquad \Gamma \vdash_v v \Rightarrow A}{\mathsf{tr}(A) \subseteq \mathsf{tr}(\Gamma, v) \ \ \textit{or}\ v = r_v} \qquad\qquad \frac{t\,\mathsf{OK} \qquad \Gamma \vdash_c t \Rightarrow X}{\mathsf{tr}(X) \subseteq \mathsf{tr}(\Gamma, t) \ \ \textit{or}\ t = r_c}$$

*Proof.* Mutual induction. □

## 14.2 Trace Transferred from Context

As the declarative type formation inference rules are intertwined with typing and subtyping, the inductive proof forces us to augment the proved invariant. Specifically, the invariant we are proving is spread across the following three lemmas. The proof should be read as one mutual induction.

**Lemma 18** (The Safe Variable Invariant for Type Formation)**.**

$$\frac{\Gamma[\hat{x} : A] \vdash X\ \mathsf{ctype} \qquad \hat{x} \in^{!} X\ \mathsf{OK}}{\mathsf{tr}(\Gamma, A) \subseteq \mathsf{tr}(\Gamma, X)} \qquad\qquad \frac{\Gamma[\hat{x} : A] \vdash B\ \mathsf{vtype} \qquad \hat{x} \in^{!} B\ \mathsf{OK}}{\mathsf{tr}(\Gamma, A) \subseteq \mathsf{tr}(\Gamma, B)}$$

**Lemma 19** (The Safe Variable Invariant for Typing)**.**

$$\frac{v\,\mathsf{OK} \qquad \Gamma[\hat{x} : A] \vdash_v v \Leftarrow C \qquad \hat{x} \in^{!} v\,\mathsf{OK}}{\mathsf{tr}(\Gamma, A) \subseteq \mathsf{tr}(\Gamma, v) \cup \mathsf{tr}(\Gamma, C)} \qquad\qquad \frac{e\,\mathsf{OK} \qquad \Gamma[\hat{x} : A] \vdash_v e \Leftarrow Y \qquad \hat{x} \in^{!} e\,\mathsf{OK}}{\mathsf{tr}(\Gamma, A) \subseteq \mathsf{tr}(\Gamma, e) \cup \mathsf{tr}(\Gamma, Y)}$$

$$\frac{v\,\mathsf{OK} \qquad \Gamma[\hat{x} : A] \vdash_v v \Rightarrow C \qquad \hat{x} \in^{!} v\,\mathsf{OK}}{\mathsf{tr}(\Gamma, A) \subseteq \mathsf{tr}(\Gamma, v) \cup \mathsf{tr}(\Gamma, C)} \qquad\qquad \frac{e\,\mathsf{OK} \qquad \Gamma[\hat{x} : A] \vdash_v e \Rightarrow Y \qquad \hat{x} \in^{!} e\,\mathsf{OK}}{\mathsf{tr}(\Gamma, A) \subseteq \mathsf{tr}(\Gamma, e) \cup \mathsf{tr}(\Gamma, Y)}$$

**Lemma 20** (The Safe Variable Invariant for Subtyping)**.**

$$\frac{\Gamma \vdash A \leqslant^{v} B}{\mathsf{tr}(\Gamma, A) = \mathsf{tr}(\Gamma, B)} \qquad\qquad \frac{\Gamma \vdash X \leqslant^{c} Y}{\mathsf{tr}(\Gamma, X) \subseteq \mathsf{tr}(\Gamma, Y)}$$

*Proof of lemma 18.* Mutual (with lemmas 19 and 20) induction on the type formation judgement.

**VarT** Holds vacuously, since $\hat{x} \notin^{!} a$.

□

*Proof of lemma 20.* Mutual (with lemmas 18 and 19) induction on the subtyping judgement.

$\leqslant$**Var** Trivial.

$\leqslant \Sigma$ By the induction hypothesis and congruence of $\mathsf{EV}()$ and $\Downarrow$.

$\leqslant \Pi$ Similar to the $\leqslant \Sigma$ case. It is essential that the proved property is symmetrical for the value-subtyping.

$\leqslant \mathcal{U}$ By the induction hypothesis. The symmetry of the premise of the rule is essential.

$\leqslant \mathcal{F}$ By the induction hypothesis. The symmetry of the premise is <u>not</u> essential here.

$\leqslant$**Eq** By the induction hypothesis and lemma 11.

$\leqslant \forall$ $\mathsf{tr}(X) \subseteq \mathsf{tr}(Y) \subseteq \mathsf{tr}(\forall y : A.\, Y)$.

**let$\leqslant$ and $\leqslant$let** After one evaluation step, the proven inclusion coincides with the induction hypothesis.

**let$\leqslant$let** By congruence, the induction hypothesis, and lemma 11.

$\forall \leqslant$

$$\begin{aligned}
&\mathsf{tr}(\forall x : A.\, X) && \text{(by congruence)}\\
\subseteq\ &\mathsf{tr}(A) \cup \mathsf{tr}(X) && \text{(by lemma 18, because}\\
& && \Gamma, x : A \vdash X\ \mathsf{ctype})\\
\subseteq\ &\mathsf{tr}(X)\\
\subseteq\ &\mathsf{EV}((\Downarrow X)\{x := \Downarrow v\}) && \text{(by corollary 1)}\\
=\ &\mathsf{tr}((X\{x := v\})) && \text{(by the IH)}\\
\subseteq\ &\mathsf{tr}(Y)
\end{aligned}$$

□