The Proofs

ANONYMOUS AUTHOR(S)

ACM Reference Format:
Anonymous Author(s). 2024. The Proofs. 1, 1 (February 2024), 120 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Contents

1 DECLARATIVE TYPE SYSTEMS

1.1 Grammar

We assume that there is an infinite set of positive and negative *type* variables. Positive type variables are denoted as α^+, β^+, γ^+, etc. Negative type variables are denoted as α^-, β^-, γ^-, etc. We assume there is an infinite set of *term* variables, which are denoted as x, y, z, etc. A list of objects (variables, types or terms) is denoted by an overline arrow. For instance, $\overrightarrow{\alpha^+}$ is a list of positive type variables, $\overrightarrow{\beta^-}$ is a list of negative type variables, $\overrightarrow{v}$ is a list of values, which are arguments of a function. **fv** (P) and **fv** (N) denote the set of free variables in a type P and N, respectively.

Definition 1 (Declarative Types).

Negative declarative types

$$N,\ M,\ K \quad ::= \quad \mid\ \alpha^- \quad \mid\ \uparrow P \quad \mid\ P \to N \quad \mid\ \forall \overrightarrow{\alpha^+}.\ N$$

Positive declarative types

$$P,\ Q,\ R \quad ::= \quad \mid\ \alpha^+ \quad \mid\ \downarrow N \quad \mid\ \exists \overrightarrow{\alpha^-}.\ P$$

1.2 Equalities

For simplicity, we assume alpha-equivalent terms are equal. This way, we assume that substitutions do not capture bound variables. Besides, we equate $\forall \overrightarrow{\alpha^+}.\ \forall \overrightarrow{\beta^+}.\ N$ with $\forall \overrightarrow{\alpha^+}, \overrightarrow{\beta^+}.\ N$, as well as $\exists \overrightarrow{\alpha^-}.\ \exists \overrightarrow{\beta^-}.\ P$ with $\exists \overrightarrow{\alpha^-}, \overrightarrow{\beta^-}.\ P$, and lift these equations transitively and congruently to the whole system.

1.3 Contexts and Well-formedness

Definition 2 (Declarative Type Context).
Declarative type context T is represented by a set of type variables. The concatenation T_1, T_2 means the union of two contexts $T_1 \cup T_2$.

$T \vdash P$ and $T \vdash N$ denote that the type is well-formed in the context T, which means that each free type variable of the type is contained in T (it will be shown later in lemmas 1 and 2).

Notice that checking the well-formedness of a type is an *algorithmic* procedure, in which both the context and the type are considered inputs. In other words, it is syntax-directed and mode-correct (according to [**dunfieldBidirectionalTyping2020**]), which means that checking the well-formedness of a type can be done recursively by a deterministic algorithm. We will use the well-formedness checking in the inference algorithm, for example, to check that the existential variables do not escape their scope.

Algorithm 1 (Type Well-formedness).

$\boxed{T \vdash N}$ *Negative type well-formedness*

$$\frac{\alpha^- \in T}{T \vdash \alpha^-}\ (\textsc{Var}_-^{WF})$$

$$\frac{T \vdash P}{T \vdash \uparrow P}\ (\uparrow^{WF})$$

$$\frac{T \vdash P \quad T \vdash N}{T \vdash P \to N}\ (\to^{WF})$$

$$\frac{T, \overrightarrow{\alpha^+} \vdash N}{T \vdash \forall \overrightarrow{\alpha^+}.\ N}\ (\forall^{WF})$$

$\boxed{T \vdash P}$ *Positive type well-formedness*

$$\frac{\alpha^+ \in T}{T \vdash \alpha^+} \; (\text{VAR}_+^{WF}) \qquad \frac{T \vdash N}{T \vdash \downarrow N} \; (\downarrow^{WF}) \qquad \frac{T, \overrightarrow{\alpha^-} \vdash P}{T \vdash \exists \overrightarrow{\alpha^-}.\, P} \; (\exists^{WF})$$

1.4 Substitutions

Definition 3 (Substitution). *Substitutions (denoted as σ) are represented by total functions from variables to types, preserving the polarity.*

Algorithm 2 (Substitution Application). *Substitution application is denoted as $[\sigma]P$ and $[\sigma]N$. It is defined naturally as follows:*

$[\sigma]\alpha^+ = \sigma(\alpha^+)$
$[\sigma]\alpha^- = \sigma(\alpha^-)$
$[\sigma]\downarrow N = \downarrow[\sigma]N$
$[\sigma]\uparrow P = \uparrow[\sigma]P$
$[\sigma](P \to N) = [\sigma]P \to [\sigma]N$
$[\sigma]\exists\overrightarrow{\alpha^-}.\, Q = \exists\overrightarrow{\alpha^-}.\, [\sigma]Q$
$[\sigma]\forall\overrightarrow{\alpha^+}.\, N = \forall\overrightarrow{\alpha^+}.\, [\sigma]N$ *(assuming the variable capture never happens)*

Definition 4 (Substitution Signature). *The signature $T' \vdash \sigma : T$ means that*

(1) for any $\alpha^\pm \in T, T' \vdash [\sigma]\alpha^\pm$; and
(2) for any $\alpha^\pm \notin T', [\sigma]\alpha^\pm = \alpha^\pm$.

A substitution can be restricted to a set of variables. The restricted substitution is define as expected.

Definition 5 (Subsitution Restriction). *The specification $\sigma|_{vars}$ is defined as a function such that*

(1) $\sigma|_{vars}(\alpha^\pm) = \sigma(\alpha^\pm)$, if $\alpha^\pm \in vars$; and
(2) $\sigma|_{vars}(\alpha^\pm) = \alpha^\pm$, if $\alpha^\pm \notin vars$.

Two substitutions can be composed in two ways: $\sigma_2 \circ \sigma_1$ corresponds to a consecutive application of σ_1 and σ_2, while $\sigma_2 \lll \sigma_1$ depends on a signature of σ_1 and modifies σ_1 by applying σ_2 to its results on the domain.

Definition 6 (Substitution Composition). *$\sigma_2 \circ \sigma_1$ is defined as a function such that $\sigma_2 \circ \sigma_1(\alpha^\pm) = \sigma_2(\sigma_1(\alpha^\pm))$.*

Definition 7 (Monadic Substitution Composition). *Suppose that $T' \vdash \sigma_1 : T$. Then we define $\sigma_2 \lll \sigma_1$ as $(\sigma_2 \circ \sigma_1)|_T$.*

Notice that the result of $\sigma_2 \lll \sigma_1$ depends on the specification of σ_1, which is not unique. However, we assume that the used specification clear from the context of the proof.

Definition 8 (Equivalent Substitutions). *The substitution equivalence judgement $T' \vdash \sigma_1 \simeq^{\leqslant} \sigma_2 : T$ indicates that on the domain T, the result of σ_1 and σ_2 are equivalent in context T'. Formally, for any $\alpha^\pm \in T, T' \vdash [\sigma_1]\alpha^\pm \simeq^{\leqslant} [\sigma_2]\alpha^\pm$.*

Sometimes it is convenient to construct substitution explicitly mapping each variable from a list (or a set) to a type. Such substitutions are denoted as $\overrightarrow{P}/\overrightarrow{\alpha^+}$ and $\overrightarrow{N}/\overrightarrow{\alpha^-}$, where $\overrightarrow{P}$ and $\overrightarrow{N}$ are lists of the corresponding types.

Definition 9 (Explicit Substitution).

– *Suppose that $\overrightarrow{\alpha^-}$ is a list of negative type variables, and $\overrightarrow{N}$ is a list of negative types of the same length. Then $\overrightarrow{N}/\overrightarrow{\alpha^-}$ denotes a substitution such that*

(1) for $\alpha_i^+ \in \overrightarrow{\alpha^-}$, $[\overrightarrow{N}/\overrightarrow{\alpha^-}]\alpha_i^+ = N_i$;
(2) for $\beta^+ \notin \overrightarrow{\alpha^-}$, $[\overrightarrow{N}/\overrightarrow{\alpha^-}]\beta^+ = \beta^+$.

+ *Positive explicit substitution* $\overrightarrow{P}/\overrightarrow{\alpha^+}$ *is defined symmetrically.*

1.5 Declarative Subtyping

Subtyping is one of the key mechanisms of our system. It realizes the polymorphism: abstract ∀ and ∃ types can be used where concrete types are expected, exactly because of the subtyping relation between them.

Definition 10.

$\boxed{T \vdash N \leqslant M}$ *Negative subtyping*

$$\frac{}{T \vdash \alpha^- \leqslant \alpha^-}\ (\text{VAR}_-^{\leqslant})$$

$$\frac{T \vdash P \simeq^{\leqslant} Q}{T \vdash \uparrow P \leqslant \uparrow Q}\ (\uparrow^{\leqslant})$$

$$\frac{T \vdash P \geqslant Q \quad T \vdash N \leqslant M}{T \vdash P \to N \leqslant Q \to M}\ (\to^{\leqslant})$$

$$\frac{T, \overrightarrow{\beta^+} \vdash \sigma : \overrightarrow{\alpha^+} \quad T, \overrightarrow{\beta^+} \vdash [\sigma]N \leqslant M}{T \vdash \forall \overrightarrow{\alpha^+}.\, N \leqslant \forall \overrightarrow{\beta^+}.\, M}\ (\forall^{\leqslant})$$

$\boxed{T \vdash P \geqslant Q}$ *Positive supertyping*

$$\frac{}{T \vdash \alpha^+ \geqslant \alpha^+}\ (\text{VAR}_+^{\geqslant})$$

$$\frac{T \vdash N \simeq^{\leqslant} M}{T \vdash \downarrow N \geqslant \downarrow M}\ (\downarrow^{\geqslant})$$

$$\frac{T, \overrightarrow{\beta^-} \vdash \sigma : \overrightarrow{\alpha^-} \quad T, \overrightarrow{\beta^-} \vdash [\sigma]P \geqslant Q}{T \vdash \exists \overrightarrow{\alpha^-}.\, P \geqslant \exists \overrightarrow{\beta^-}.\, Q}\ (\exists^{\geqslant})$$

$\boxed{T \vdash N \simeq^{\leqslant} M}$ *Negative equivalence*

$$\frac{T \vdash N \leqslant M \quad T \vdash M \leqslant N}{T \vdash N \simeq^{\leqslant} M}\ (\simeq_-^{\leqslant})$$

$\boxed{T \vdash P \simeq^{\leqslant} Q}$ *Positive equivalence*

$$\frac{T \vdash P \geqslant Q \quad T \vdash Q \geqslant P}{T \vdash P \simeq^{\leqslant} Q}\ (\simeq_+^{\leqslant})$$

The following observations about the declarative subtyping are worth noting:

- ($\text{VAR}_-^{\leqslant}$) and ($\text{VAR}_+^{\geqslant}$) make the subtyping reflexive on variables (and further, on any type).
- ($\to^{\leqslant}$) is standard: the arrow is covariant on the resulting type and contravariant on the argument type.
- ($\downarrow^{\geqslant}$) and ($\uparrow^{\leqslant}$) are non-standard: the subtyping is *invariant* for shifts. This way, the subtyping of shifted types in one direction implies the subtyping in the opposite direction. Although this rule restricts the subtyping relation, it makes the system decidable.
- ($\forall^{\leqslant}$) and ($\exists^{\geqslant}$) are the only non-algorithmic rules: the substitution for the quantified variable is not specified, those, these rules 'drive' the subtyping relation.

In the next section, we present the sound and complete algorithm checking whether one type is a subtype of another according to definition 10.

2 ALGORITHMIC TYPE SYSTEM

2.1 Grammar

In the algorithmic system, we extend the grammar of types by adding positive and negative *algorithmic variables* ($\widehat{\alpha}^+$, $\widehat{\beta}^+$, $\widehat{\gamma}^+$, etc. and $\widehat{\alpha}^-$, $\widehat{\beta}^-$, $\widehat{\gamma}^-$, etc.). They represent the unknown types, which will be inferred by the algorithm. This way, we add two base cases to the grammar of positive and negative types and use highlight to denote that the type can potentially contain algorithmic variables.

Definition 11 (Algorithmic Types).

Negative algorithmic type			*Positive algorithmic type*		
N, M	$::=$		P, Q	$::=$	
	$\mid$	$\widehat{\alpha}^-$		$\mid$	$\widehat{\alpha}^+$
	$\mid$	α^-		$\mid$	α^+
	$\mid$	$\uparrow P$		$\mid$	$\downarrow N$
	$\mid$	$P \to N$		$\mid$	$\exists \overrightarrow{\alpha^-}. P$
	$\mid$	$\forall \overrightarrow{\alpha^+}. N$			

2.2 Fresh Variable Selection

Both the subtyping and the type inference algorithm rely on the ability to select fresh, unused variables. For a set of variables *vars*, it is indicated as *vars* **are fresh** in the inference rules. We assume that the selection subroutine always succeeds and is deterministic. In other words, whenever it is called in an algorithmic inference rule, it returns the same result, uniquely determined by the input of this rule.

2.3 Variable Algorithmization

In several places of our algorithm, in particular, during algorithmic subtyping, we turn a declarative type into an algorithmic one via replacing certain type variables with fresh algorithmic variables. We call this procedure *variable algorithmization*, and define it as follows.

Definition 12 (Variable Algorithmization). *Suppose that $\overrightarrow{\alpha^-}$ is a list of negative type variables and $\overrightarrow{\widehat{\alpha}^-}$ is a list of negative algorithmic variables of the same length. Then $\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}$ is a substitution-like procedure replacing each $\alpha_i^- \in \overrightarrow{\alpha^-}$ in a type for $\widehat{\alpha}_i^- \in \overrightarrow{\widehat{\alpha}^-}$.*

Conversely, we have the opposite procedure turning algorithmic type variables into declarative type variables via *dealgorithmization*.

Definition 13 (Variable Dealgorithmization). *Suppose that $\overrightarrow{\widehat{\alpha}^-}$ is a list of negative algorithmic variables and $\overrightarrow{\alpha^-}$ is a list of negative type variables of the same length. Then $\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}$ is a substitution-like procedure replacing each $\widehat{\alpha}_i^- \in \overrightarrow{\widehat{\alpha}^-}$ in a type for $\alpha_i^- \in \overrightarrow{\alpha^-}$.*

2.4 Contexts and Well-formedness

Definition 14 (Algorithmic Type Context Υ).
Algorithmic type context Υ is represented by a set of algorithmic *type variables ($\widehat{\alpha}^+, \widehat{\alpha}^-, \widehat{\beta}^+, \dots$). The concatenation Υ_1, Υ_2 means the union of two contexts $\Upsilon_1 \cup \Upsilon_2$.*

$T ; \Upsilon \vdash P$ and $T ; \Upsilon \vdash N$ are used to denote that the algorithmic type is well-formed in the contexts T and Υ, which means that each algorithmic variable of the type is contained in Υ, and each free declarative type variable of the type is contained in T.

Algorithm 3 (Algorithmic Type Well-formedness).
$\boxed{T ; \Upsilon \vdash N}$ *Negative type well-formedness*

$$\frac{\widehat{\alpha}^- \in \Upsilon}{T ; \Upsilon \vdash \widehat{\alpha}^-} \quad (\text{UVAR}_-^{WF})$$

$$\frac{\alpha^- \in T}{T ; \Upsilon \vdash \alpha^-} \quad (\text{VAR}_-^{WF})$$

$$\frac{T ; \Upsilon \vdash P}{T ; \Upsilon \vdash \uparrow P} \quad (\uparrow^{WF})$$

$\boxed{T\,;\Upsilon \vdash P}$ *Positive type well-formedness*

$$\frac{T\,;\Upsilon \vdash P \quad T\,;\Upsilon \vdash N}{T\,;\Upsilon \vdash P \rightarrow N}\ (\rightarrow^{WF})$$

$$\frac{\alpha^+ \in T}{T\,;\Upsilon \vdash \alpha^+}\ (\mathrm{VAR}_+^{WF})$$

$$\frac{\widehat{\alpha}^+ \in \Upsilon}{T\,;\Upsilon \vdash \widehat{\alpha}^+}\ (\mathrm{UVAR}_+^{WF})$$

$$\frac{T\,;\Upsilon \vdash N}{T\,;\Upsilon \vdash \downarrow N}\ (\downarrow^{WF})$$

$$\frac{T, \overrightarrow{\alpha^+}\,;\Upsilon \vdash N}{T\,;\Upsilon \vdash \forall\overrightarrow{\alpha^+}.\,N}\ (\forall^{WF})$$

$$\frac{T, \overrightarrow{\alpha^-}\,;\Upsilon \vdash P}{T\,;\Upsilon \vdash \exists\overrightarrow{\alpha^-}.\,P}\ (\exists^{WF})$$

Algorithmic Type Context are used in the unification algorithm. In the subtyping algorithm, the context needs to remember additional information. In the subtyping context, each algorithmic variable is associated with a context it must be instantiated in (i.e. the context in which the type replacing the variable must be well-formed). This association is represented by *algorithmic subtyping context* Σ.

Definition 15 (Algorithmic Subtyping Context Σ)**.**
Algorithmic Subtyping Context Σ is represented by a set of entries of form $\widehat{\alpha}^+\{T\}$ and $\widehat{\alpha}^-\{T\}$, where $\widehat{\alpha}^+$ and $\widehat{\alpha}^-$ are algorithmic variables, and T is a context in which they must be instantiated. We assume that no two entries associating the same variable appear in Σ.

dom (Σ) *denotes the set of variables appearing in Σ:* **dom** $(\Sigma) = \{\widehat{\alpha}^\pm \mid \widehat{\alpha}^\pm\{T\} \in \Sigma\}$. *If $\widehat{\alpha}^\pm\{T\} \in \Sigma$, we denote T as $\Sigma(\widehat{\alpha}^\pm)$.*

2.5 Subsitutions

Substitution that operates on algorithmic type variables is denoted as $\widehat{\sigma}$. It is defined as a total function from algorithmic type variables to *declarative* types, preserving the polarity.

The signature $\Sigma \vdash \widehat{\sigma} : \Upsilon$ means that $\Upsilon \subseteq$ **dom** (Σ) and $\widehat{\sigma}$ maps each algorithmic variable from Υ to a type well-formed in $\Sigma(\widehat{\alpha}^\pm)$; and for each variable not appearing in **dom** (Σ), it acts as identity.

Definition 16 (Signature of Algorithmic Substitution)**.**

- $\Sigma \vdash \widehat{\sigma} : \Upsilon$ *means that*
  (1) for any $\widehat{\alpha}^\pm \in \Upsilon$, there exists T such that $\widehat{\alpha}^\pm\{T\} \in \Sigma$ and $T \vdash [\widehat{\sigma}]\widehat{\alpha}^\pm$;
  (2) for any $\widehat{\alpha}^\pm \notin \Upsilon$, $[\widehat{\sigma}]\widehat{\alpha}^\pm = \widehat{\alpha}^\pm$.
- $T \vdash \widehat{\sigma} : \Upsilon$ *means that*
  (1) for any $\widehat{\alpha}^\pm \in \Upsilon$, $T \vdash [\widehat{\sigma}]\widehat{\alpha}^\pm$;
  (2) for any $\widehat{\alpha}^\pm \notin \Upsilon$, $[\widehat{\sigma}]\widehat{\alpha}^\pm = \widehat{\alpha}^\pm$.

In the anti-unification algorithm, we use another kind of substitution. In contrast to algorithmic substitution $\widehat{\sigma}$, it allows mapping algorithmic variables to *algorithmic* types. Additionally, anti-unification substitution is restricted to the *negative* segment of the language. Anti-unification substitution is denoted as $\widehat{\tau}$ and $\widehat{\rho}$.a

The pair of contexts T and Υ, in which the results of an anti-unification substitution are formed, is fixed for this substitution. This way, $T\,;\Upsilon_2 \vdash \widehat{\tau} : \Upsilon_1$ means that $\widehat{\tau}$ maps each negative algorithmic variable appearing in Υ_1 to a term well-formed in T and Υ_2.

Definition 17 (Signature of Anti-unification substitution)**.** *$T\,;\Upsilon_2 \vdash \widehat{\tau} : \Upsilon_1$ means that*

(1) for any $\widehat{\alpha}^- \in \Upsilon_1$, $T ; \Upsilon_2 \vdash [\widehat{\tau}]\widehat{\alpha}^-$ and
(2) for any $\widehat{\alpha}^- \notin \Upsilon_1$, $[\widehat{\tau}]\widehat{\alpha}^- = \widehat{\alpha}^-$.

2.6 Equivalence and Normalization

The subtyping-induced equivalence (definition 10) is non-trivial: there are types that are subtypes of each other but not equal. For example, $\forall \alpha^+, \beta^+.\ \alpha^+ \to \uparrow\beta^+$ is a subtype and a supertype of $\forall \alpha^+, \beta^+.\ \beta^+ \to \uparrow\alpha^+$ and of, for example, $\forall \alpha^+, \beta^+.\ \beta^+ \to \uparrow\exists\gamma^-.\ \alpha^+$, although these types are not alpha-equivalent. For the subtyping algorithm, it is crucial to be able to check whether two types are equivalent, without checking mutual subtyping. For this purpose we define the normalization procedure, which allows us to uniformly choose the representative type of the equivalence class. This way, the equivalence checking is reduced to normalization and equality checking.

For clarification of the proofs and better understanding of the system, we introduce an intermediate relation—*declarative equivalence*. As will be shown in lemmas 27 and 32, this relation is equivalent to the subtyping-induced equivalence, but does not depend on it. Although this relation is not defined algorithmically, it gives the intuition of what types our system considers equivalent. Specifically, in addition to *alpha-equivalence*, our system allows for *reordering of adjacent quantifiers*, and *introduction/elimination of unused quantifiers*.

The non-trivial rules of the declarative equivalence are ($\forall^{\simeq^D}$) and ($\exists^{\simeq^D}$). Intuitively, the variable bijection μ reorders the quantifiers before the recursive call on the body of the quantified type. It will be covered formally in section 8.4.

Definition 18 (Declarative Type Equivalence).
$\boxed{N \simeq^D M}$ *Negative type equivalence* $\boxed{P \simeq^D Q}$ *Positive type equivalence*

$$\frac{}{\alpha^- \simeq^D \alpha^-}\ (\textsc{Var}^{\simeq^D}_-) \qquad \frac{}{\alpha^+ \simeq^D \alpha^+}\ (\textsc{Var}^{\simeq^D}_+)$$

$$\frac{P \simeq^D Q}{\uparrow P \simeq^D \uparrow Q}\ (\uparrow^{\simeq^D}) \qquad \frac{N \simeq^D M}{\downarrow N \simeq^D \downarrow M}\ (\downarrow^{\simeq^D})$$

$$\frac{P \simeq^D Q \quad N \simeq^D M}{P \to N \simeq^D Q \to M}\ (\to^{\simeq^D})$$

$$\frac{\begin{array}{c}\mu : (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M) \leftrightarrow (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N) \\ \overrightarrow{\alpha^+} \cap \mathbf{fv}\, M = \emptyset \quad N \simeq^D [\mu]M\end{array}}{\forall \overrightarrow{\alpha^+}.\ N \simeq^D \forall \overrightarrow{\beta^+}.\ M}\ (\forall^{\simeq^D}) \qquad \frac{\begin{array}{c}\mu : (\overrightarrow{\beta^-} \cap \mathbf{fv}\, Q) \leftrightarrow (\overrightarrow{\alpha^-} \cap \mathbf{fv}\, P) \\ \overrightarrow{\alpha^-} \cap \mathbf{fv}\, Q = \emptyset \quad P \simeq^D [\mu]Q\end{array}}{\exists \overrightarrow{\alpha^-}.\ P \simeq^D \exists \overrightarrow{\beta^-}.\ Q}\ (\exists^{\simeq^D})$$

As the equivalence includes arbitrary reordering of quantified variables, the normalization procedure is needed to choose the canonical order. For this purpose, we introduce an auxiliary procedure—variable ordering. Intuitively, **ord** *vars* **in** N returns a list of variables from *vars* in the order they appear in N.

Algorithm 4 (Variable Ordering).
$\boxed{\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ N = \vec{\alpha}}$ *variable ordering in a negative type*

$$\frac{\alpha^- \in \mathit{vars}}{\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ \alpha^- = \alpha^-}\ (\textsc{Var}^{\textsc{Ord}}_{-\in})$$

$$\frac{\alpha^- \notin \mathit{vars}}{\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ \alpha^- = \cdot}\ (\textsc{Var}^{\textsc{Ord}}_{-\notin})$$

$$\frac{\textbf{ord } vars \textbf{ in } P = \vec{\alpha}}{\textbf{ord } vars \textbf{ in } \uparrow P = \vec{\alpha}} \quad (\uparrow^{\text{ORD}})$$

$$\frac{\textbf{ord } vars \textbf{ in } P = \vec{\alpha}_1 \quad \textbf{ord } vars \textbf{ in } N = \vec{\alpha}_2}{\textbf{ord } vars \textbf{ in } P \to N = \vec{\alpha}_1, (\vec{\alpha}_2 \setminus \vec{\alpha}_1)} \quad (\to^{\text{ORD}})$$

$$\frac{vars \cap \overrightarrow{\alpha^+} = \emptyset \quad \textbf{ord } vars \textbf{ in } N = \vec{\alpha}}{\textbf{ord } vars \textbf{ in } \forall \overrightarrow{\alpha^+}.\, N = \vec{\alpha}} \quad (\forall^{\text{ORD}})$$

$\boxed{\textbf{ord } vars \textbf{ in } P = \vec{\alpha}}$
variable ordering in a positive type

$$\frac{\alpha^+ \in vars}{\textbf{ord } vars \textbf{ in } \alpha^+ = \alpha^+} \quad (\text{VAR}_{+\in}^{\text{ORD}})$$

$$\frac{\alpha^+ \notin vars}{\textbf{ord } vars \textbf{ in } \alpha^+ = \cdot} \quad (\text{VAR}_{+\notin}^{\text{ORD}})$$

$$\frac{\textbf{ord } vars \textbf{ in } N = \vec{\alpha}}{\textbf{ord } vars \textbf{ in } \downarrow N = \vec{\alpha}} \quad (\downarrow^{\text{ORD}})$$

$$\frac{vars \cap \overrightarrow{\alpha^-} = \emptyset \quad \textbf{ord } vars \textbf{ in } P = \vec{\alpha}}{\textbf{ord } vars \textbf{ in } \exists \overrightarrow{\alpha^-}.\, P = \vec{\alpha}} \quad (\exists^{\text{ORD}})$$

Analogously, the variable can be ordered in an algorithmic *type* (**ord** *vars* **in** P *and* **ord** *vars* **in** N)*. In these cases, we treat the algorithmic variables as if they were declarative variables.*

Next, we use the variable ordering in the normalization procedure. Specifically, normalization recursively traverses the type, and for each quantified case reorders the quantified variables in a canonical order dictated by algorithm 4, removing unused ones.

Algorithm 5 (Type Normalization)**.**

$\boxed{\textbf{nf } (N) = M}$

$$\frac{}{\textbf{nf } (\alpha^-) = \alpha^-} \quad (\text{VAR}_{-}^{\text{NF}})$$

$$\frac{\textbf{nf } (P) = Q}{\textbf{nf } (\uparrow P) = \uparrow Q} \quad (\uparrow^{\text{NF}})$$

$$\frac{\textbf{nf } (P) = Q \quad \textbf{nf } (N) = M}{\textbf{nf } (P \to N) = Q \to M} \quad (\to^{\text{NF}})$$

$$\frac{\textbf{nf } (N) = N' \quad \textbf{ord } \overrightarrow{\alpha^+} \textbf{ in } N' = \overrightarrow{\alpha^+}'}{\textbf{nf } (\forall \overrightarrow{\alpha^+}.\, N) = \forall \overrightarrow{\alpha^+}'.\, N'} \quad (\forall^{\text{NF}})$$

$\boxed{\textbf{nf } (P) = Q}$

$$\frac{}{\textbf{nf } (\alpha^+) = \alpha^+} \quad (\text{VAR}_{+}^{\text{NF}})$$

$$\frac{\textbf{nf } (N) = M}{\textbf{nf } (\downarrow N) = \downarrow M} \quad (\downarrow^{\text{NF}})$$

$$\frac{\textbf{nf } (P) = P' \quad \textbf{ord } \overrightarrow{\alpha^-} \textbf{ in } P' = \overrightarrow{\alpha^-}'}{\textbf{nf } (\exists \overrightarrow{\alpha^-}.\, P) = \exists \overrightarrow{\alpha^-}'.\, P'} \quad (\exists^{\text{NF}})$$

Analogously, we define the normalization of algorithmic types by adding base cases:

$\boxed{\mathbf{nf}\ (N) = M}$ $\boxed{\mathbf{nf}\ (P) = Q}$

$$\frac{}{\mathbf{nf}\ (\widehat{\alpha}^-) = \widehat{\alpha}^-}\ (\textit{UVar}^{\textit{NF}}_-) \qquad \frac{}{\mathbf{nf}\ (\widehat{\alpha}^+) = \widehat{\alpha}^+}\ (\textit{UVar}^{\textit{NF}}_+)$$

Lemma 46 demonstrates that the equivalence of types is the same as the equality of their normal forms.

Theorem (Correctness of Normalization). *Assuming the types are well-formed in T,*

- $T \vdash N \simeq^{\leqslant} M$ *if and only if* $\mathbf{nf}\ (N) = \mathbf{nf}\ (M)$;

+ $T \vdash P \simeq^{\leqslant} Q$ *if and only if* $\mathbf{nf}\ (P) = \mathbf{nf}\ (Q)$.

Algorithm 6 (Substitution Normalization). *For a substitution σ, we define* $\mathbf{nf}\ (\sigma)$ *as a substitution that maps $\alpha^{\pm}$ into* $\mathbf{nf}\ ([\sigma]\alpha^{\pm})$.

The rest of this chapter is devoted to the central algorithm of the type system—the subtyping algorithm. Figure 1 shows the dependency graph of the subtyping algorithm. The nodes represent the algorithmic procedures, and the edge $A \longrightarrow B$ means that A uses B as a sub-procedure.

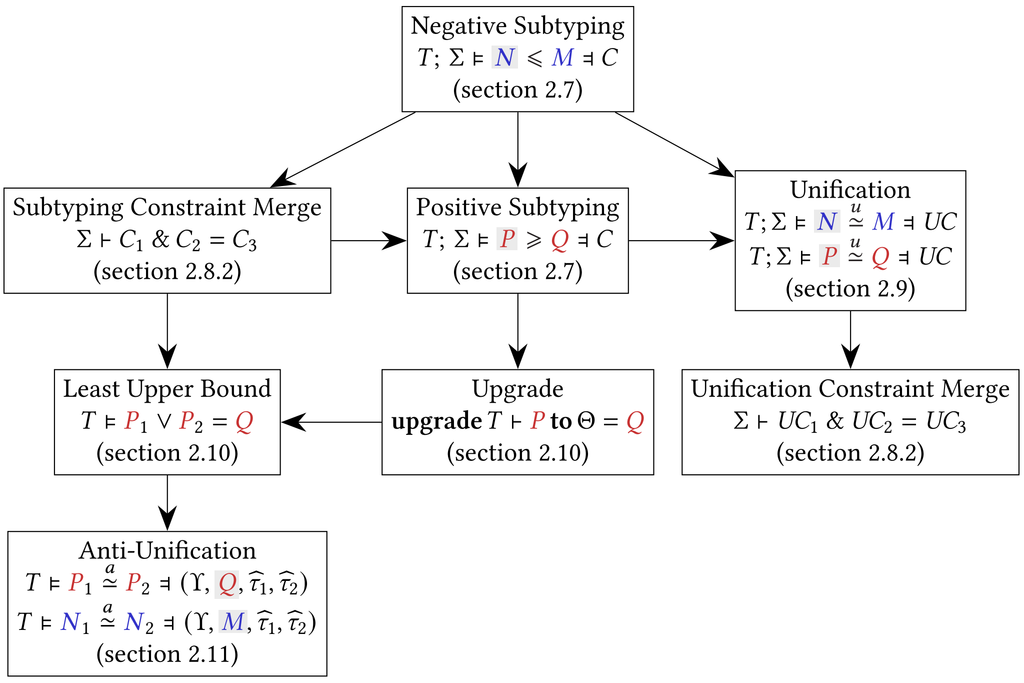
Negative Subtyping
$T;\ \Sigma \vDash N \leqslant M \dashv C$
(section 2.7)
Subtyping Constraint Merge
$\Sigma \vdash C_1\ \&\ C_2 = C_3$
(section 2.8.2)
Positive Subtyping
$T;\ \Sigma \vDash P \geqslant Q \dashv C$
(section 2.7)
Unification
$T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC$
$T; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$
(section 2.9)
Least Upper Bound
$T \vDash P_1 \vee P_2 = Q$
(section 2.10)
Upgrade
upgrade $T \vdash P$ **to** $\Theta = Q$
(section 2.10)
Unification Constraint Merge
$\Sigma \vdash UC_1\ \&\ UC_2 = UC_3$
(section 2.8.2)
Anti-Unification
$T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$
$T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$
(section 2.11)

Fig. 1. Dependency graph of the subtyping algorithm

2.7 Subtyping

Now, we present the subtyping algorithm itself. Although the algorithm is presented as a single procedure, is important for the structure of the proof that the positive subtyping algorithm does not invoke the negative one. This way, the correctness of the positive subtyping will be proved independently and used afterwards to prove the correctness of the negative subtyping.

Algorithm 7 (Subtyping).

$\boxed{T;\Sigma \vDash N \leqslant M \dashv C}$ *Negative subtyping*

$$\frac{}{T;\ \Sigma \vDash \alpha^{-} \leqslant \alpha^{-} \dashv \cdot}\ (\textsc{Var}^{\leqslant}_{-})$$

$$\frac{T;\Sigma \vDash \mathbf{nf}\,(P) \overset{u}{\simeq} \mathbf{nf}\,(Q) \dashv UC}{T;\ \Sigma \vDash \uparrow P \leqslant \uparrow Q \dashv UC}\ (\uparrow^{\leqslant})$$

$$\frac{\begin{array}{l}\overrightarrow{\widehat{\alpha}^{+}}\ \textbf{are fresh}\\ T, \overrightarrow{\beta^{+}};\ \Sigma, \overrightarrow{\widehat{\alpha}^{+}}\{T, \overrightarrow{\beta^{+}}\} \vDash [\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N \leqslant M \dashv C\end{array}}{T;\ \Sigma \vDash \forall\overrightarrow{\alpha^{+}}.\ N \leqslant \forall\overrightarrow{\beta^{+}}.\ M \dashv C \setminus \overrightarrow{\widehat{\alpha}^{+}}}\ (\forall^{\leqslant})$$

$$\frac{\begin{array}{l}T;\ \Sigma \vDash P \geqslant Q \dashv C_1 \quad T;\ \Sigma \vDash N \leqslant M \dashv C_2\\ \Sigma \vdash C_1 \,\&\, C_2 = C\end{array}}{T;\ \Sigma \vDash P \to N \leqslant Q \to M \dashv C}\ (\to^{\leqslant})$$

$\boxed{T;\Sigma \vDash P \geqslant Q \dashv C}$ *Positive supertyping*

$$\frac{}{T;\ \Sigma \vDash \alpha^{+} \geqslant \alpha^{+} \dashv \cdot}\ (\textsc{Var}^{\geqslant}_{+})$$

$$\frac{\begin{array}{l}\overrightarrow{\widehat{\alpha}^{-}}\ \textbf{are fresh}\\ T, \overrightarrow{\beta^{-}};\ \Sigma, \overrightarrow{\widehat{\alpha}^{-}}\{T, \overrightarrow{\beta^{-}}\} \vDash [\overrightarrow{\widehat{\alpha}^{-}}/\overrightarrow{\alpha^{-}}]P \geqslant Q \dashv C\end{array}}{T;\ \Sigma \vDash \exists\overrightarrow{\alpha^{-}}.\ P \geqslant \exists\overrightarrow{\beta^{-}}.\ Q \dashv C \setminus \overrightarrow{\widehat{\alpha}^{-}}}\ (\exists^{\geqslant})$$

$$\frac{T;\Sigma \vDash \mathbf{nf}\,(N) \overset{u}{\simeq} \mathbf{nf}\,(M) \dashv UC}{T;\ \Sigma \vDash \downarrow N \geqslant \downarrow M \dashv UC}\ (\downarrow^{\geqslant})$$

$$\frac{\mathbf{upgrade}\ T \vdash P\ \mathbf{to}\ \Sigma(\widehat{\alpha}^{+}) = Q}{T;\ \Sigma \vDash \widehat{\alpha}^{+} \geqslant P \dashv (\widehat{\alpha}^{+} :\geqslant Q)}\ (\textsc{UVar}^{\geqslant})$$

The inputs of the subtyping algorithm are the declarative context T, the subtyping context Σ (it specifies in which contexts the algorithmic variables must be instantiated), and the types themselves: N and M for the negative case, and P and Q for the positive case. As one of the invariants, we require M and Q to be declarative (i.e. not containing algorithmic variables). The output of the algorithm is a set of *subtyping constraints* C, which will be discussed in the next section.

Let us overview the inference rules of the subtyping algorithm.

- ($\textsc{Var}^{\leqslant}_{-}$) and ($\textsc{Var}^{\geqslant}_{+}$) are the base cases. They copy the corresponding declarative rules and ensure reflexivity.
- ($\textsc{UVar}^{\geqslant}$) is the only case generating subtyping constraints. In this case, we must ensure that the resulting constraints guarantee that the instantiation of $\widehat{\alpha}^{+}$ is a supertype of P. However, the obvious constraint $\widehat{\alpha}^{+} :\geqslant P$ might be problematic if P is not well-formed in $\Sigma(\widehat{\alpha}^{+})$. For this reason, we use the *upgrade* procedure (it will be covered in section 2.10) to find the minimal supertype of P, which is well-formed in $\Sigma(\widehat{\alpha}^{+})$.
  Notice that this rule does not have a negative counterpart. This is because one of the important invariants of the algorithm: in the negative subtyping, only positive algorithmic variables can occur in the types.

- ($\downarrow^{\geqslant}$) and ($\uparrow^{\leqslant}$) are the *shift* rules. According to the declarative system, shifted subtyping requires equivalence. In the presence of the algorithmic variables, it means that the left and the right-hand sides of the subtyping must be unified. Hence, the shift rules invoke the unification algorithm, which will be discussed in section 2.9. The unification returns the minimal set of constraints UC, which is necessary and sufficient for the subtyping.
- ($\rightarrow^{\leqslant}$). In this case, the algorithm makes two calls: a recursive call to the negative subtyping algorithm for the argument types, and a call to the positive subtyping algorithm for the result types. After that, the resulting constraints are merged using the *subtyping constraint merge* procedure, which is discussed in section 2.8.2.
- ($\forall^{\leqslant}$) and ($\exists^{\geqslant}$) are symmetric. These are the only places where the algorithmic variables are introduced. It is done by algorithmization (section 2.3) of the quantified variables: these variables are replaced by fresh algorithmic variables in the body of the quantified type, the algorithmic variables are added to the subtyping context Σ, after that, the recursive call is made. Notice that the declarative context T is extended by the quantified variables from the right-hand side, which matches the declarative system.

Then soundness lemma (lemmas 78 and 84) and completeness (lemmas 79 and 85) of the algorithm together give us the following simplified theorem:

Theorem (Correctness of subtyping algorithm)**.**

$-$ $T; \cdot \vDash N \leqslant M \dashv \cdot$ *is equivalent to* $T \vdash N \leqslant M$;
$+$ $T; \cdot \vDash P \geqslant Q \dashv \cdot$ *is equivalent to* $T \vdash P \geqslant Q$.

2.8 Constraints

Unification and subtyping algorithms are based on constraint generation. The constraints are represented by a set of constraint entries.

Definition 19 (Unification Constraint)**.**

unification entry *(denoted as ue) is an expression of shape* $\widehat{\alpha}^+ :\simeq P$ *or* $\widehat{\alpha}^- :\simeq N$;
unification constraint *(denoted as UC) is a set of unification constraint entries. We denote* $\{\widehat{\alpha}^\pm \mid ue \in UC \text{ restricting } \widehat{\alpha}^\pm\}$ *as* **dom** (UC).

However, in the subtyping, we need to consider more general kind of constraints. Specifically, subtyping constraint entries can restrict a variable not only to be equivalent to a certain type, but also to be a supertype of a positive type.

Definition 20 (Subtyping Constraint)**.**

subtyping entry *(denoted as e) is an expression of shape* $\widehat{\alpha}^+ :\geqslant P$, $\widehat{\alpha}^- :\simeq N$, *or* $\widehat{\alpha}^+ :\simeq P$;
subtyping constraint *(denoted as C) is a set of subtyping constraint entries. We denote* $\{\widehat{\alpha}^\pm \mid e \in C \text{ restricting } \widehat{\alpha}^\pm\}$ *as* **dom** (C).

Definition 21 (Well-formed Constraint Entry)**.** *We say that a constraint entry is well-formed in a context T if its associated type is well-formed in T.*

$T \vdash \widehat{\alpha}^+ :\geqslant P$ *iff* $T \vdash P$;
$T \vdash \widehat{\alpha}^+ :\simeq P$ *iff* $T \vdash P$;
$T \vdash \widehat{\alpha}^- :\simeq N$ *iff* $T \vdash N$.

Definition 22 (Well-formed Constraint)**.** *We say that a constraint is well-formed in a subtyping context* Σ *if all its entries are well-formed in the corresponding elements of* Σ*. More formally,* $\Sigma \vdash C$ *holds iff for every* $e \in C$*, such that e restricts* $\widehat{\alpha}^\pm$*, we have* $\Sigma(\widehat{\alpha}^\pm) \vdash e$.

We write $\Sigma \vdash C : \Upsilon$ *to denote that* $\Sigma \vdash C$ *and* **dom** $(C) = \Upsilon$.

$\Sigma \vdash UC$ *and* $\Sigma \vdash UC : \Upsilon$ *are defined analogously.*

2.8.1 Constraint Satisfaction. A constraint entry restricts a type that can be assigned to a variable. We say that a type satisfies a constraint entry if it can be assigned to the variable restricted by the entry.

Definition 23 (Type Satisfying a Constraint Entry).

$\boxed{T \vdash N : e}$ *Negative entry satisfaction* $\boxed{T \vdash P : e}$ *Positive entry satisfaction*

$$\frac{T \vdash N \simeq^{\leqslant} M}{T \vdash N : (\widehat{\alpha}^{-} :\simeq M)} \quad (:\simeq^{SAT}_{-})$$

$$\frac{T \vdash P \geqslant Q}{T \vdash P : (\widehat{\alpha}^{+} :\geqslant Q)} \quad (:\geqslant^{SAT}_{+})$$

$$\frac{T \vdash P \simeq^{\leqslant} Q}{T \vdash P : (\widehat{\alpha}^{+} :\simeq Q)} \quad (:\simeq^{SAT}_{+})$$

We say that a substitution satisfies a constraint—a set of constraint entries if each entry is satisfied by the type assigned to the variable by the substitution.

Definition 24 (Substitution Satisfying a Constraint). *We write $\Sigma \vdash \widehat{\sigma} : C$ to denote that a substitution $\widehat{\sigma}$ satisfies a constraint C in a context Σ. It presumes that $\Sigma \vdash C$ and means that for any $ue \in C$, if ue restricts $\widehat{\alpha}^{\pm}$, then $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}]\widehat{\alpha}^{\pm} : ue$.*

Unification constraint satisfaction $\Sigma \vdash \widehat{\sigma} : UC$ is defined analogously as a special case of subtyping constraint satisfaction.

Notice that $\Sigma \vdash \widehat{\sigma} : C$ does not imply the signature $\Sigma \vdash \widehat{\sigma} : \mathbf{dom}\,(C)$, because the latter also specifies $\widehat{\sigma}$ outside of the domain $\mathbf{dom}\,(C)$ (see definition 16).

2.8.2 Constraint Merge. In this section, define the least upper bound for constraints, which we call *merge*. Intuitively, the merge of two constraints is the least constraint such that any substitution satisfying both constraints satisfies the merge as well. First, we define the merge of entries, and then extend it to the set of entries.

Definition 25 (Matching Entries). *We call two unification constraint entries or two subtyping constraint entries matching if they are restricting the same unification variable.*

Two matching entries formed in the same context T can be merged in the following way:

Algorithm 8 (Merge of Matching Constraint Entries).
$\boxed{T \vdash e_1 \,\&\, e_2 = e_3}$ *Subtyping Constraint Entry Merge*

$$\frac{T \vDash P_1 \vee P_2 = Q}{T \vdash (\widehat{\alpha}^{+} :\geqslant P_1) \,\&\, (\widehat{\alpha}^{+} :\geqslant P_2) = (\widehat{\alpha}^{+} :\geqslant Q)} \quad (\geqslant \&^{+} \geqslant)$$

$$\frac{T;\, \cdot \vDash P \geqslant Q \dashv \cdot}{T \vdash (\widehat{\alpha}^{+} :\simeq P) \,\&\, (\widehat{\alpha}^{+} :\geqslant Q) = (\widehat{\alpha}^{+} :\simeq P)} \quad (\simeq \&^{+} \geqslant)$$

$$\frac{T;\, \cdot \vDash Q \geqslant P \dashv \cdot}{T \vdash (\widehat{\alpha}^{+} :\geqslant P) \,\&\, (\widehat{\alpha}^{+} :\simeq Q) = (\widehat{\alpha}^{+} :\simeq Q)} \quad (\geqslant \&^{+} \simeq)$$

$$\frac{\mathbf{nf}\,(P) = \mathbf{nf}\,(P')}{T \vdash (\widehat{\alpha}^{+} :\simeq P) \,\&\, (\widehat{\alpha}^{+} :\simeq P') = (\widehat{\alpha}^{+} :\simeq P)} \quad (\simeq \&^{+} \simeq)$$

$$\frac{\mathbf{nf}\,(N) = \mathbf{nf}\,(N')}{T \vdash (\widehat{\alpha}^- :\simeq N)\ \&\ (\widehat{\alpha}^- :\simeq N') = (\widehat{\alpha}^- :\simeq N)} \quad (\simeq \&^- \simeq)$$

- $(\simeq \&^+ \simeq)$ and $(\simeq \&^- \simeq)$ are symmetric cases. To merge two matching entries restricting a variable to be equivalent to certain types, we check that these types are equivalent to each other. To do so, it suffices to check for *equality* of their normal forms, as discussed in section 2.6. After that, we return the left-hand entry.
- $(\simeq \&^+ \geqslant)$ and $(\geqslant \&^+ \simeq)$ are also symmetric. In this case, since one of the entries requires the variable to be equal to a type, the resulting entry must also imply that. However, for the soundness, it is needed to ensure that the equating restriction is stronger than the subtyping restriction. For this purpose, the premise invokes the positive subtyping.
- $(\geqslant \&^+ \geqslant)$ In this case, we find the least upper bound of the types from the input restrictions, and as the output, restrict the variable to be a supertype of the result. The least upper bound procedure will be discussed in section 2.10.

Unification constraint entries are a special case of subtyping constraint entries. They are merged using the same algorithm (algorithm 8). Notice that the merge of two matching unification constraint entries is a unification constraint entry.

Lemma 1 (Merge of Matching Unification Constraint Entries is well-defined). *Suppose that $T \vdash ue_1$ and $T \vdash ue_2$ are unification constraint entries. Then the merge of ue_1 and ue_2 $T \vdash ue_1 \,\&\, ue_2 = ue$ according to algorithm 8, is a unification constraint entry.*

Proof. Since ue_1 and ue_2 are matching unification constraint entries, they have the shape $(\widehat{\alpha}^+ :\simeq P_1, \widehat{\alpha}^+ :\simeq P_2)$ or $(\widehat{\alpha}^- :\simeq N_1, \widehat{\alpha}^- :\simeq N_2)$. Then the merge of ue_1 and ue_2 can only be defined by $(\simeq \&^+ \simeq)$ or $(\simeq \&^- \simeq)$. In both cases the result, if it exists, is a unification constraint entry: in the first case, the result has shape $\widehat{\alpha}^+ :\simeq P_1$, in the second case, the result has shape $\widehat{\alpha}^- :\simeq N_1$. □

Algorithm 9 (Merge of Subtyping Constraints). *Suppose that $\Sigma \vdash C_1$ and $\Sigma \vdash C_2$. Then $\Sigma \vdash C_1 \,\&\, C_2 = C$ defines a set of constraints C such that $e \in C$ iff either:*

- *$e \in C_1$ and there is no matching $e' \in C_2$; or*
- *$e \in C_2$ and there is no matching $e' \in C_1$; or*
- *$\Sigma(\widehat{\alpha}^\pm) \vdash e_1 \,\&\, e_2 = e$ for some $e_1 \in C_1$ and $e_2 \in C_2$ such that e_1 and e_2 both restrict variable $\widehat{\alpha}^\pm$.*

Unification constraints can be considered as a special case of subtyping constraints, and the merge of unification constraints is defined as the merge of subtyping constraints. Then it is easy to see that the merge of two unification constraints is a unification constraint.

Lemma 2 (Merge of Unification Constraints is well-defined). *Suppose that $\Sigma \vdash UC_1$ and $\Sigma \vdash UC_2$ are unification constraints. Then the merge of UC_1 and UC_2 $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$ according to algorithm 9, is a unification constraint.*

Proof. UC consists of unmatched entries of UC_1 and UC_2, which are *unification* constraint entries by assumption, and merge of matching entries, which also are *unification* constraint entries by lemma 1. □

Lemmas 81 and 83 show the correctness and initiality of the merge operation, which can be expressed in the following simplified theorem:

Theorem (Correctness of Constraint Merge). *A substitution $\widehat{\sigma}$ satisfying both constraints C_1 and C_2 if and only if it satisfies their merge.*

The unification constraint merge satisfies the same theorem, however, because the merge of unification constraint entries ue_1 and ue_2 always results in one of them, a stronger soundness property holds (see lemma 61):

Theorem (Soundness of Unification Constraint Merge). *If* $\Sigma \vdash UC_1 \mathbin{\&} UC_2 = UC$ *then* $UC = UC_1 \cup UC_2$.

2.9 Unification

The subtyping algorithm calls the following subtask: given two algorithmic types, we need to find the most general substitution for the algorithmic variables in these types, such that the resulting types are equivalent. This problem is known as *unification.*

In our case, the unification is restricted in the following way: first, before unifying the types, we normalize them, which allows us to reduce (non-trivial) equivalence to (trivial) equality; second, we preserve invariants which guarantee that one side of the unification is always declarative, which in fact, reduces the unification to the *matching* problem.

The unification procedure returns a set of minimal constraints, that must be satisfied by a substitution unifying the input types.

Algorithm 10 (Unification).

$\boxed{T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC}$ *Negative unification*

$\boxed{T; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC}$ *Positive unification*

$$\frac{}{T; \Sigma \vDash \alpha^- \overset{u}{\simeq} \alpha^- \dashv \cdot}\ (\text{VAR}^{\overset{u}{\simeq}}_-) \qquad \frac{}{T; \Sigma \vDash \alpha^+ \overset{u}{\simeq} \alpha^+ \dashv \cdot}\ (\text{VAR}^{\overset{u}{\simeq}}_+)$$

$$\frac{T; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC}{T; \Sigma \vDash \uparrow P \overset{u}{\simeq} \uparrow Q \dashv UC}\ (\uparrow^{\overset{u}{\simeq}}) \qquad \frac{T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC}{T; \Sigma \vDash \downarrow N \overset{u}{\simeq} \downarrow M \dashv UC}\ (\downarrow^{\overset{u}{\simeq}})$$

$$\frac{T; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC_1 \quad T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC_2}{T; \Sigma \vDash P \to N \overset{u}{\simeq} Q \to M \dashv UC_1 \mathbin{\&} UC_2}\ (\to^{\overset{u}{\simeq}})$$

$$\frac{T, \overrightarrow{\alpha^-}; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC}{T; \Sigma \vDash \exists \overrightarrow{\alpha^-}.\, P \overset{u}{\simeq} \exists \overrightarrow{\alpha^-}.\, Q \dashv UC}\ (\exists^{\overset{u}{\simeq}})$$

$$\frac{T, \overrightarrow{\alpha^+}; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC}{T; \Sigma \vDash \forall \overrightarrow{\alpha^+}.\, N \overset{u}{\simeq} \forall \overrightarrow{\alpha^+}.\, M \dashv UC}\ (\forall^{\overset{u}{\simeq}})$$

$$\frac{\Sigma(\widehat{\alpha}^-) \vdash N}{T; \Sigma \vDash \widehat{\alpha}^- \overset{u}{\simeq} N \dashv (\widehat{\alpha}^- :\simeq N)}\ (\text{UVAR}^{\overset{u}{\simeq}}_-) \qquad \frac{\Sigma(\widehat{\alpha}^+) \vdash P}{T; \Sigma \vDash \widehat{\alpha}^+ \overset{u}{\simeq} P \dashv (\widehat{\alpha}^+ :\simeq P)}\ (\text{UVAR}^{\overset{u}{\simeq}}_+)$$

- $(\uparrow^{\overset{u}{\simeq}})$, $(\downarrow^{\overset{u}{\simeq}})$, $(\forall^{\overset{u}{\simeq}})$, and $(\exists^{\overset{u}{\simeq}})$ are defined congruently. In the shift rules, the algorithm removes the outermost constructor. In the $\forall$ and $\exists$ rules, it removes the quantifiers, adding the quantified variables to the context T. Notice that Σ, which specifies the contexts in which the algorithmic variables must be instantiated, is not changed.
- $(\text{VAR}^{\overset{u}{\simeq}}_-)$ and $(\text{VAR}^{\overset{u}{\simeq}}_+)$ are the base cases. Since the sides are equal and free from algorithmic variables, the unification returns an empty constraint.
- $(\text{VAR}^{\overset{u}{\simeq}}_-)$ and $(\text{VAR}^{\overset{u}{\simeq}}_+)$ are symmetric cases constructing the constraints. When an algorithmic variable is unified with a type, we must check that the type is well-formed in the required context, and if it is, we return a constraint restricting the variable to be equivalent to that type.
- $(\to^{\overset{u}{\simeq}})$. In this case, the algorithm makes two recursive calls: it unifies the arguments and the results of the arrows. After that, the resulting constraints are merged using the *unification*

constraint merge procedure, which is discussed in section 2.8.2. Notice that UC_1 and UC_2 are guaranteed to be *unification* constraints, not arbitrary *subtyping* constraints: it is important for modularizing the proofs, since the properties of the *unification* constraint merge can be proved independently from the *subtyping* constraint merge.

2.10 Least Upper Bound

In this section, we present the algorithm finding the least common supertype of two positive types. It is used directly by the constraint merge procedure (section 2.8.2), and indirectly, through the type upgrade by positive subtyping (section 2.7). Perhaps, the least upper bound is the least intuitive part of the algorithm, and its correctness will be covered in section 9.8.

Algorithm 11 (The Least Upper Bound Algorithm)**.**
$\boxed{T \vDash P_1 \vee P_2 = Q}$ *Least Upper Bound*

$$\frac{T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-} \vDash P_1 \vee P_2 = Q}{T \vDash \exists \overrightarrow{\alpha^-}.\, P_1 \vee \exists \overrightarrow{\beta^-}.\, P_2 = Q} \quad (\exists^{\vee})$$

$$\frac{}{T \vDash \alpha^+ \vee \alpha^+ = \alpha^+} \quad (\text{VAR}^{\vee})$$

$$\frac{T \vDash \mathbf{nf}\,(\downarrow N) \overset{a}{\simeq} \mathbf{nf}\,(\downarrow M) \dashv (\Upsilon, P, \widehat{\tau}_1, \widehat{\tau}_2)}{T \vDash \downarrow N \vee \downarrow M = \exists \overrightarrow{\alpha^-}.\, [\overrightarrow{\alpha^-}/\Upsilon]\, P} \quad (\downarrow^{\vee})$$

- ($\text{VAR}^{\vee}$) The base case is trivial: the least upper bound of to equal variables is the variable itself.
- ($\downarrow^{\vee}$) In case both sides of the least upper bound are shifted, the algorithm needs to find the anti-unifier of them. Intuitively, this is because in general, the upper bounds of $\downarrow N$ are $\exists \overrightarrow{\alpha^-}.\, P$ such that $\overrightarrow{\alpha^-}$ can be instantiated with some $\overrightarrow{M}$ so that $T \vdash [\overrightarrow{M}/\overrightarrow{\alpha^-}]P \simeq^{\leqslant} \downarrow N$ (see lemma 69).
- ($\exists^{\vee}$) In this case, we move the quantified variables to the context T, and make a recursive call. It is important to make sure that $\overrightarrow{\alpha^-}$ and $\overrightarrow{\beta^-}$ are disjoint. In this case, it is guaranteed that the resulting $\mathbf{fv}\,(Q)$ will be free of $\overrightarrow{\alpha^-}$ and $\overrightarrow{\beta^-}$, and thus, the resulting type will be a supertype of both sides (it will be discussed in lemma 69).

In the positive subtyping algorithm (section 2.7), ($\text{UVAR}^{\geqslant}$) generates a restriction of a variable $\widehat{\alpha}^+$. On the one hand, this restriction must imply $\widehat{\alpha}^+ :\geqslant P$ for the subtyping to hold. On the other hand, the type used in this restriction must be well-formed in a potentially stronger (smaller) context than P.

To resolve this problem, we define the *upgrade* procedure, which for given $\Theta, \overrightarrow{\alpha^{\pm}}$, and $\Theta, \overrightarrow{\alpha^{\pm}} \vdash P$, finds $\Theta \vdash Q$—the least supertype of P among the types well-formed in Θ.

The trick is to make sure that the 'forbidden' variables $\overrightarrow{\alpha^{\pm}}$ are not used explicitly in the supertypes of P. For this purpose, we construct new types P_1 and P_2, in each of them replacing the forbidden variables with fresh variables $\overrightarrow{\beta^{\pm}}$ and $\overrightarrow{\gamma^{\pm}}$, and then find the least upper bound of P_1 and P_2. It turns out that this renaming forces the common types of P_1 and P_2 to be agnostic to $\overrightarrow{\alpha^{\pm}}$, and thus, the supertypes of P well-formed in Θ are exactly the common supertypes of P_1 and P_2. These properties are considered in more details in section 9.9.

Algorithm 12 (Type Upgrade)**.**
$\boxed{\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q}$ *Type Upgrade*

$$\frac{\begin{array}{l} T = \Theta, \overrightarrow{\alpha^{\pm}} \\ \overrightarrow{\beta^{\pm}} \textbf{ are fresh} \quad \overrightarrow{\gamma^{\pm}} \textbf{ are fresh} \\ \Theta, \overrightarrow{\beta^{\pm}}, \overrightarrow{\gamma^{\pm}} \vDash [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P \vee [\overrightarrow{\gamma^{\pm}}/\overrightarrow{\alpha^{\pm}}]P = Q \end{array}}{\textbf{upgrade } T \vdash P \textbf{ to } \Theta = Q} \quad (\text{UPG})$$

Note on the Greatest Lower Bound. In contrast to the least upper bound, the general greatest lower bound does not exist in our system. For instance, consider a positive type P, together with its non-equivalent supertypes P_1 and $P_2 \not\simeq P_1$ (for example, $P = \downarrow\uparrow\downarrow\gamma^-$, $P_1 = \exists\alpha^-. \downarrow\uparrow\downarrow\alpha^-$, and $P_2 = \exists\alpha^-. \downarrow\alpha^-$). Then for arbitrary Q and N, let us consider the common subtypes of $A = Q \to \downarrow\uparrow Q \to \downarrow\uparrow Q \to N$ and $B = P \to \downarrow\uparrow P_1 \to \downarrow\uparrow P_2 \to N$. It is easy to see that $\forall\alpha^+. \forall\beta^+. \alpha^+ \to \downarrow\uparrow\alpha^+ \to \downarrow\uparrow\beta^+ \to N$ and $\forall\alpha^+. \forall\beta^+. \alpha^+ \to \downarrow\uparrow\beta^+ \to \downarrow\uparrow\alpha^+ \to N$ are both *maximal* common subtypes of A and B, and since they are not equivalent, none of them is the *greatest* one.

However, we designed the subtyping system in such a way that the greatest lower bound is not needed: the negative variables are always 'protected' by *invariant* shifts ($\uparrow$ and $\downarrow$), and thus, the algorithm can only require a substitution of a negative variable to be *equivalent* to some type but never to be a *subtype*.

2.11 Anti-unification

Next, we define the anti-unification procedure, also known as the *most specific generalization*. As an input, it takes two declarative types (e.g., in the positive case P_1 and P_2) and a context T. and returns a type Q—the generalizer, containing negative placeholders (represented by algorithmic variables) from Υ and two substitutions $\widehat{\tau}_1$ and $\widehat{\tau}_2$. The substitutions replace the placeholders with declarative types well-formed in T, such that $[\widehat{\tau}_1]Q = P_1$ and $[\widehat{\tau}_2]Q = P_2$. Moreover, the algorithm guarantees that Q is the most specific type with this property: any other generalizer can be turned into Q by some substitution $\widehat{\rho}$.

It is important to note the differences between the standard anti-unification and our version. First, we only allow the placeholders at *negative* positions, which means, for example, that α^+ and β^+ cannot be generalized. Second, the generated pair of substitutions $\widehat{\tau}_1$ and $\widehat{\tau}_2$ must replace the placeholders with types well-formed in a specified context T.

The anti-unification algorithm assumes that the input types are normalized. This way, anti-unification up-to-equality rather than anti-unification up-to-equivalence is sufficient.

Algorithm 13 (Anti-unification).

$$\boxed{T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)}$$

$$\frac{}{T \vDash \alpha^+ \overset{a}{\simeq} \alpha^+ \dashv (\cdot, \alpha^+, \cdot, \cdot)} \quad (\text{VAR}^{\overset{a}{\simeq}}_+)$$

$$\frac{T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)}{T \vDash \downarrow N_1 \overset{a}{\simeq} \downarrow N_2 \dashv (\Upsilon, \downarrow M, \widehat{\tau}_1, \widehat{\tau}_2)} \quad (\downarrow^{\overset{a}{\simeq}})$$

$$\frac{\overrightarrow{\alpha^-} \cap T = \emptyset \quad T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)}{T \vDash \exists\overrightarrow{\alpha^-}. P_1 \overset{a}{\simeq} \exists\overrightarrow{\alpha^-}. P_2 \dashv (\Upsilon, \exists\overrightarrow{\alpha^-}. Q, \widehat{\tau}_1, \widehat{\tau}_2)} \quad (\exists^{\overset{a}{\simeq}})$$

$$\boxed{T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)}$$

$$\frac{}{T \vDash \alpha^- \overset{a}{\simeq} \alpha^- \dashv (\cdot, \alpha^-, \cdot, \cdot)}\quad (\textit{V}\textsc{ar}^{\overset{a}{\simeq}}_-)$$

$$\frac{T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)}{T \vDash \uparrow P_1 \overset{a}{\simeq} \uparrow P_2 \dashv (\Upsilon, \uparrow Q, \widehat{\tau}_1, \widehat{\tau}_2)}\quad (\uparrow^{\overset{a}{\simeq}})$$

$$\frac{\overrightarrow{\alpha^+} \cap T = \emptyset \quad T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)}{T \vDash \forall \overrightarrow{\alpha^+}.\, N_1 \overset{a}{\simeq} \forall \overrightarrow{\alpha^+}.\, N_2 \dashv (\Upsilon, \forall \overrightarrow{\alpha^+}.\, M, \widehat{\tau}_1, \widehat{\tau}_2)}\quad (\forall^{\overset{a}{\simeq}})$$

$$\frac{T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon_1, Q, \widehat{\tau}_1, \widehat{\tau}_2) \quad T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon_2, M, \widehat{\tau}'_1, \widehat{\tau}'_2)}{T \vDash P_1 \to N_1 \overset{a}{\simeq} P_2 \to N_2 \dashv (\Upsilon_1 \cup \Upsilon_2, Q \to M, \widehat{\tau}_1 \cup \widehat{\tau}'_1, \widehat{\tau}_2 \cup \widehat{\tau}'_2)}\quad (\to^{\overset{a}{\simeq}})$$

$$\frac{\textit{if other rules are not applicable} \quad T \vdash N \quad T \vdash M}{T \vDash N \overset{a}{\simeq} M \dashv (\widehat{\alpha}^-_{\{N,M\}}, \widehat{\alpha}^-_{\{N,M\}}, (\widehat{\alpha}^-_{\{N,M\}} \mapsto N), (\widehat{\alpha}^-_{\{N,M\}} \mapsto M))}\quad (\textit{AU})$$

- ($\textsc{Var}^{\overset{a}{\simeq}}_+$) and ($\textsc{Var}^{\overset{a}{\simeq}}_-$) are the base cases. In this case, since the input types are equal, the algorithm returns this type as a generalizer, without generating any placeholders.
- ($\downarrow^{\overset{a}{\simeq}}$), ($\uparrow^{\overset{a}{\simeq}}$), ($\forall^{\overset{a}{\simeq}}$), and ($\exists^{\overset{a}{\simeq}}$) are defined congruently. In the shift rules, the algorithm removes the outermost constructor. In the $\forall$ and $\exists$ rules, it removes the quantifiers. Notice that the algorithm does not add the removed variables to the context T. This is because T is used to restrict the resulting anti-unification substitutions, and is fixed throughout the algorithm.
- (AU) is the most important rule, since it generates the placeholders. This rule only applies if other negative rules failed. Because of that, the anti-unification procedure is *not* syntax-directed.
  The generated placeholder is indexed with a pair of types it is mapped to. It allows the algorithm to automatically unite the anti-unification solutions generated by the different branches of ($\to^{\overset{a}{\simeq}}$).
  Notice that this rule does not have a positive counterpart, since we only allow negative placeholders.
- ($\to^{\overset{a}{\simeq}}$) makes two recursive calls to the anti-unification procedure, and unites the results. Suppose that $\widehat{\tau}_1$ and $\widehat{\tau}_2$ are the substitutions generated by anti-unification of *argument* types of the arrow, and $\widehat{\tau}'_1$ and $\widehat{\tau}'_2$ are the substitutions generated by anti-unification of *result* types of the arrow. It is important that if ($\widehat{\tau}_1$, $\widehat{\tau}_2$) and ($\widehat{\tau}'_1$, $\widehat{\tau}'_2$) send some variables to the same pair of types, i.e., $[\widehat{\tau}_1]\widehat{\alpha}^- = [\widehat{\tau}'_1]\widehat{\beta}^-$ and $[\widehat{\tau}_2]\widehat{\alpha}^- = [\widehat{\tau}'_2]\widehat{\beta}^-$, then these variables are equal, i.e., $\widehat{\alpha}^- = \widehat{\beta}^-$. This property is guaranteed by (AU): the name of the placeholder is determined by the pair of types it is mapped to.

3 DECLARATIVE TYPING

In the previous section, we presented the type system together with subtyping specification and the algorithm. In this section, we describe the language under this type system, together with the type inference specification and algorithm.

3.1 Grammar

First, we define the syntax of the language. The language combines System F with call-by-push-value style.

Definition 26 (Language Grammar)**.**

Computation Terms			*Value Terms*		
$c,\ d$	$::=$		$v,\ w$	$::=$	
	$\mid$	$(c : N)$		$\mid$	x
	$\mid$	$\lambda x : P.\ c$		$\mid$	$\{c\}$
	$\mid$	$\Lambda\alpha^{+}.\ c$		$\mid$	$(v : P)$
	$\mid$	$\textbf{return}\ v$			
	$\mid$	$\textbf{let}\ x = v\ ;\ c$			
	$\mid$	$\textbf{let}\ x : P = v(\overrightarrow{v})\ ;\ c$			
	$\mid$	$\textbf{let}\ x = v(\overrightarrow{v})\ ;\ c$			
	$\mid$	$\textbf{let}^{\exists}(\overrightarrow{\alpha^{-}}, x) = v\ ;\ c$			

Notice that the language does not have first-class applications: instead, we use applicative let bindings— constructions that bind a result of a fully applied function to a (positive) variable. In the call-by-push-value paradigm, it corresponds to monadic bind or do-notation. Typewise, these let-binders come in two forms: annotated and unannotated. The annotated let-binders $\textbf{let}\ x : P = v(\overrightarrow{v}); c$ requires the application to infer the annotated P, whereas the unannotated $\textbf{let}\ x = v(\overrightarrow{v})\ ;\ c$ is used when the inferred type is unique.

A computation of a polymorphic type is constructed using $\Lambda\alpha^{+}.\ c$, however, the elimination of $\forall$ is implicit. Conversely, the existential types are constructed implicitly and eliminated using the standard unpack mechanism: $\textbf{let}^{\exists}(\overrightarrow{\alpha^{-}}, x) = v\ ;\ c$.

Another dual pair of constructions are $\textbf{return}\ v$ and $\{c\}$. The former allows us to embed a value in pure computations. The latter, on the contrary, encapsulates a thunk of computation in a value.

Finally, the language has several standard constructions: lambda-abstractions $\lambda x : P.\ c$, standard let-bindings $\textbf{let}\ x = v\ ;\ c$, and type annotations that can be added to any value or computation: $(v : P)$ and $(c : N)$.

3.2 Declarative Type Inference

Next, we define the specification of the type inference for our language. First, we introduce variable context specifying the types of variables in the scope of the current rule.

Definition 27 (Variable Context)**.** *The variable typing context Γ is represented by a set of entries of the form $x : P$.*

The specification is represented by an inference system of three mutually recursive judgments: positive inference $T; \Gamma \vdash v \colon P$, negative type inference $T; \Gamma \vdash c \colon N$, and application type inference $T\ ; \Gamma \vdash N \bullet \overrightarrow{v} \Rrightarrow M$. In the premises, the inference rules also refer to the declarative subtyping (definition 10), type well-formedness (algorithm 1), and normalization (algorithm 5).

- $T;\Gamma \vdash v\colon P$ (and symmetrically, $T;\Gamma \vdash c\colon N$) means that under the type context T and the variable context Γ, for the value v, type P is inferrable. It guarantees that v is well-formed in T and Γ in the standard sense.
- $T\,;\Gamma \vdash N \bullet \vec{v} \Longrightarrow M$ is the application type inference judgment. It means that if a head of type N is applied to list of values $\vec{v}$, then the resulting computation can be typed as M.

Definition 28 (Declarative Type Inference). $\boxed{T;\Gamma \vdash c\colon N}$ *Negative typing*

$$\frac{T \vdash P \quad T;\Gamma, x : P \vdash c\colon N}{T;\Gamma \vdash \lambda x : P.\ c\colon P \to N}\ (\lambda^{INF})$$

$$\frac{T \vdash P \quad T;\Gamma \vdash v\colon {\downarrow}M \quad T\,;\Gamma \vdash M \bullet \vec{v} \Longrightarrow {\uparrow}Q \quad T \vdash {\uparrow}Q \leqslant {\uparrow}P \quad T;\Gamma, x : P \vdash c\colon N}{T;\Gamma \vdash \mathbf{let}\, x : P = v(\vec{v})\ ;\, c\colon N}\ (LET^{INF}_{:@})$$

$$\frac{T, \alpha^+;\Gamma \vdash c\colon N}{T;\Gamma \vdash \Lambda\alpha^+.\ c\colon \forall\alpha^+.\ N}\ (\Lambda^{INF})$$

$$\frac{T;\Gamma \vdash v\colon P}{T;\Gamma \vdash \mathbf{return}\ v\colon {\uparrow}P}\ (RET^{INF})$$

$$\frac{T;\Gamma \vdash v\colon \exists\overrightarrow{\alpha^-}.\ P \quad \mathbf{nf}\,(\exists\overrightarrow{\alpha^-}.\ P) = \exists\overrightarrow{\alpha^-}.\ P \quad T, \overrightarrow{\alpha^-};\Gamma, x : P \vdash c\colon N \quad T \vdash N}{T;\Gamma \vdash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v\ ;\, c\colon N}\ (LET^{INF}_{\exists})$$

$$\frac{T;\Gamma \vdash v\colon P \quad T;\Gamma, x : P \vdash c\colon N}{T;\Gamma \vdash \mathbf{let}\, x = v\ ;\, c\colon N}\ (LET^{INF})$$

$$\frac{T \vdash M \quad T;\Gamma \vdash c\colon N \quad T \vdash N \leqslant M}{T;\Gamma \vdash (c : M)\colon M}\ (ANN^{INF}_{-})$$

$$\frac{T;\Gamma \vdash v\colon {\downarrow}M \quad T\,;\Gamma \vdash M \bullet \vec{v} \Longrightarrow {\uparrow}Q\ \textit{principal} \quad T;\Gamma, x : Q \vdash c\colon N}{T;\Gamma \vdash \mathbf{let}\, x = v(\vec{v})\ ;\, c\colon N}\ (LET^{INF}_{@})$$

$$\frac{T;\Gamma \vdash c\colon N \quad T \vdash N \simeq^{\leqslant} N'}{T;\Gamma \vdash c\colon N'}\ (\simeq^{INF}_{-})$$

$\boxed{T;\Gamma \vdash v\colon P}$ *Positive typing*

$$\frac{x : P \in \Gamma}{T;\Gamma \vdash x\colon P}\ (\textsc{Var}^{INF})$$

$$\frac{T \vdash Q \quad T;\Gamma \vdash v\colon P \quad T \vdash Q \geqslant P}{T;\Gamma \vdash (v : Q)\colon Q}\ (ANN^{INF}_{+})$$

$$\frac{T;\Gamma \vdash c\colon N}{T;\Gamma \vdash \{c\}\colon {\downarrow}N}\ (\{\}^{INF})$$

$$\frac{T;\Gamma \vdash v\colon P \quad T \vdash P \simeq^{\leqslant} P'}{T;\Gamma \vdash v\colon P'}\ (\simeq^{INF}_{+})$$

$\boxed{T\,;\Gamma \vdash N \bullet \vec{v} \Longrightarrow M}$ *Application typing*

$$\frac{T \vdash N \simeq^{\leqslant} N'}{T\,;\Gamma \vdash N \bullet \cdot \Longrightarrow N'}\ (\emptyset^{INF}_{\bullet\Longrightarrow})$$

$$\frac{\vec{v} \neq \cdot \quad \overrightarrow{\alpha^+} \neq \cdot \quad T \vdash \sigma : \overrightarrow{\alpha^+} \quad T\,;\Gamma \vdash [\sigma]N \bullet \vec{v} \Longrightarrow M}{T\,;\Gamma \vdash \forall\overrightarrow{\alpha^+}.\ N \bullet \vec{v} \Longrightarrow M}\ (\forall^{INF}_{\bullet\Longrightarrow})$$

$$\frac{T;\Gamma \vdash v\colon P \quad T \vdash Q \geqslant P \quad T\,;\Gamma \vdash N \bullet \vec{v} \Longrightarrow M}{T\,;\Gamma \vdash Q \to N \bullet v, \vec{v} \Longrightarrow M}\ (\to^{INF}_{\bullet\Longrightarrow})$$

Let us discuss the selected rules of the declarative system:

- ($\textsc{Var}^{\textsc{inf}}$) says that the type of a variable is inferred from the context.

- ($\{\}^{\text{INF}}$) says that the type of a thunk is inferred by shifting up the type of the contained computation. Symmetrically, ($\textsc{ret}^{\text{INF}}$) infers the type of a return by shifting down the type of the contained value.
- ($\textsc{ann}^{\text{INF}}_{+}$) and ($\textsc{ann}^{\text{INF}}_{-}$) are symmetric. They allow the inferred type to be refined by annotating it with a supertype.
- ($\simeq^{\text{INF}}_{-}$) and ($\simeq^{\text{INF}}_{+}$) mean that the declarative system allows us to infer any type from the equivalence class.
- ($\textsc{let}^{\text{INF}}_{\exists}$) is standard for existential types, and its first premise infers the existential type of the value being unpacked. It is important however that the inferred existential type is normalized. This is because there might be multiple equivalent existential types with a different order or even number of quantified variables, and to bind them, the algorithm needs to fix the canonical one.
- ($\textsc{let}^{\text{INF}}_{:@}$) allows us to accommodate the applications with annotated let-bindings. The first premise infers the type of the head of the application, which must be a thunked computation. Then if after applying it to the arguments, the resulting type can be equated to the annotated one, we infer the body of the let-binding in the context extended with the bound variable.
- ($\textsc{let}^{\text{INF}}_{@}$) is similar to ($\textsc{let}^{\text{INF}}_{:@}$), bus is used for unannotated let-bindings. In this case, we require the type application to infer the 'canonical' principal type. $T\,;\Gamma \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$ principal means that any other type Q' inferrable for the application (i.e., $T\,;\Gamma \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q'$) is greater than the principal type Q, i.e., $T \vdash Q' \geqslant Q$.

Let us discuss the rules of the application inference:

- ($\emptyset^{\text{INF}}_{\bullet\Rrightarrow}$) is the base case. If the list of arguments is empty, the inferred type is the type of the head. However, we relax this specification by allowing it to infer any other equivalent type. The relaxation of this rule is enough to guarantee this property for the whole judgement: if $T\,;\Gamma \vdash N \bullet \vec{v} \Rrightarrow M$ then $T\,;\Gamma \vdash N \bullet \vec{v} \Rrightarrow M'$ for any equivalent M'.
- ($\to^{\text{INF}}_{\bullet\Rrightarrow}$) is where the application type is inferred: if the head has an arrow type $Q \to N$, we are allowed to apply it as soon as as soon as the first argument has a type, which is a subtype of Q.
- ($\forall^{\text{INF}}_{\bullet\Rrightarrow}$) is the rule ensuring the implicit elimination of the universal quantifiers. If we are applying a polymorphic computation, we can instantiate its quantified variables with any types, which is expressed by the substitution $T \vdash \sigma : \overrightarrow{\alpha^{+}}$.

4 ALGORITHMIC TYPING

Next, we present the type inference algorithm, which is sound and complete with respect to the declarative specification (definition 28).

4.1 Algorithmic Type Inference

Mirroring the declarative typing, the algorithm is represented by an inference system of three mutually recursive judgments:

- $T;\Gamma \vDash v\colon P$ and $T;\Gamma \vDash c\colon N$ are the algorithmic versions of $T;\Gamma \vdash v\colon P$ and $T;\Gamma \vdash c\colon N$. In contrast with the declarative counterparts, they are deterministic, and guarantee that the inferred type is normalized.
- $T\,;\Gamma\,;\Sigma_1 \vDash \hat{N} \bullet \vec{v} \Rrightarrow \hat{M} \dashv \Sigma_2\,;C$ is the algorithmization of $T\,;\Gamma \vdash N \bullet \vec{v} \Rrightarrow M$. Notice that $\hat{N}$ contains algorithmic variables, which are specified by the context Σ_1. Moreover, the inferred type $\hat{M}$ is also algorithmic, and can have several non-equivalent instantiations. To accommodate that, the algorithm also returns Σ_2 and C specifying the variables used

in M: Σ_2 defines the contexts in which the variables must be instantiated, and C imposes restrictions on the variables.

As subroutines, the algorithm calls subtyping (algorithm 7), type well-formedness (algorithm 1), constraint merge (section 2.8.2), normalization (algorithm 5), and constraint singularity which will be defined later in section 4.3. It also relies on basic set operations and the ability to deterministically choose fresh variables.

Algorithm 14.

$\boxed{T;\Gamma \vDash v\colon P}$ *Positive typing*

$$\frac{x : P \in \Gamma}{T;\Gamma \vDash x\colon \mathbf{nf}\,(P)}\;(\textsc{Var}^{\mathit{INF}}) \qquad \frac{\begin{array}{c} T \vdash Q \quad T;\Gamma \vDash v\colon P \\ T;\,\cdot \vDash Q \geqslant P \dashv \cdot \end{array}}{T;\Gamma \vDash (v : Q)\colon \mathbf{nf}\,(Q)}\;(\textsc{Ann}_{+}^{\mathit{INF}}) \qquad \frac{T;\Gamma \vDash c\colon N}{T;\Gamma \vDash \{c\}\colon {\downarrow} N}\;(\{\}^{\mathit{INF}})$$

$\boxed{T;\Gamma \vDash c\colon N}$ *Negative typing*

$$\frac{\begin{array}{c} T \vdash M \quad T;\Gamma \vDash c\colon N \\ T;\,\cdot \vDash N \leqslant M \dashv \cdot \end{array}}{T;\Gamma \vDash (c : M)\colon \mathbf{nf}\,(M)}\;(\textsc{Ann}_{-}^{\mathit{INF}}) \qquad \frac{T;\Gamma \vDash v\colon P}{T;\Gamma \vDash \mathbf{return}\,v\colon {\uparrow} P}\;(\textsc{Ret}^{\mathit{INF}})$$

$$\frac{T;\Gamma \vDash v\colon P \quad T;\Gamma, x : P \vDash c\colon N}{T;\Gamma \vDash \mathbf{let}\,x = v\,;\,c\colon N}\;(\textsc{Let}^{\mathit{INF}})$$

$$\frac{T \vdash P \quad T;\Gamma, x : P \vDash c\colon N}{T;\Gamma \vDash \lambda x : P.\ c\colon \mathbf{nf}\,(P \to N)}\;(\lambda^{\mathit{INF}}) \qquad \frac{\begin{array}{c} T;\Gamma \vDash v\colon \exists\overrightarrow{\alpha^-}.\,P \\ T,\overrightarrow{\alpha^-};\Gamma, x : P \vDash c\colon N \quad T \vdash N \end{array}}{T;\Gamma \vDash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v\,;\,c\colon N}\;(\textsc{Let}_{\exists}^{\mathit{INF}})$$

$$\frac{T,\alpha^+;\Gamma \vDash c\colon N}{T;\Gamma \vDash \Lambda\alpha^+.\ c\colon \mathbf{nf}\,(\forall\alpha^+.\,N)}\;(\Lambda^{\mathit{INF}})$$

$$\frac{\begin{array}{c} T \vdash P \quad T;\Gamma \vDash v\colon {\downarrow} M \quad T\,;\Gamma\,;\cdot \vDash M \bullet \vec{v} \Rrightarrow {\uparrow} Q \dashv \Sigma\,;C_1 \\ T;\,\Sigma \vDash {\uparrow} Q \leqslant {\uparrow} P \dashv C_2 \quad \Sigma \vdash C_1 \,\&\, C_2 = C \quad T;\Gamma, x : P \vDash c\colon N \end{array}}{T;\Gamma \vDash \mathbf{let}\,x : P = v(\vec{v})\,;\,c\colon N}\;(\textsc{Let}_{:@}^{\mathit{INF}})$$

$$\frac{\begin{array}{c} T;\Gamma \vDash v\colon {\downarrow} M \quad T\,;\Gamma\,;\cdot \vDash M \bullet \vec{v} \Rrightarrow {\uparrow} Q \dashv \Sigma\,;C \\ Q \text{ is } C\text{-minimized by } \widehat{\sigma} \quad T;\Gamma, x : [\widehat{\sigma}]Q \vDash c\colon N \end{array}}{T;\Gamma \vDash \mathbf{let}\,x = v(\vec{v})\,;\,c\colon N}\;(\textsc{Let}_{@}^{\mathit{INF}})$$

$\boxed{T\,;\Gamma\,;\Sigma_1 \vDash N \bullet \vec{v} \Rrightarrow M \dashv \Sigma_2\,;C}$ *Application typing*

$$\frac{}{T\,;\Gamma\,;\Sigma \vDash N \bullet \cdot \Rrightarrow \mathbf{nf}\,(N) \dashv \Sigma\,;\cdot}\;(\emptyset_{\bullet\Rrightarrow}^{\mathit{INF}}) \qquad \frac{\begin{array}{l} T;\Gamma \vDash v\colon P \quad T;\,\Sigma \vDash Q \geqslant P \dashv C_1 \\ T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M \dashv \Sigma'\,;C_2 \\ \Sigma \vdash C_1 \,\&\, C_2 = C \end{array}}{T\,;\Gamma\,;\Sigma \vDash Q \to N \bullet v, \vec{v} \Rrightarrow M \dashv \Sigma'\,;C}\;(\to_{\bullet\Rrightarrow}^{\mathit{INF}})$$

$$\frac{\begin{array}{l} T\;;\Gamma\;;\Sigma,\overrightarrow{\widehat{\alpha}^{+}}\{T\}\vDash[\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]\,N\bullet\vec{v}\Rrightarrow M\dashv\Sigma'\;;C \\ \overrightarrow{\widehat{\alpha}^{+}}\ \textbf{are fresh}\quad \vec{v}\neq\cdot\quad \overrightarrow{\alpha^{+}}\neq\cdot \end{array}}{T\;;\Gamma\;;\Sigma\vDash\forall\overrightarrow{\alpha^{+}}.\,N\bullet\vec{v}\Rrightarrow M\dashv\Sigma'\;;C|_{\mathbf{fav}(N)\cup\mathbf{fav}(M)}}\quad(\forall^{INF}_{\bullet\Rrightarrow})$$

Let us discuss the inference rules of the algorithm:

- (VAR^{INF}) infers the type of a variable by looking it up in the context and normalizing the result.
- ($\{\}^{\text{INF}}$) and (RET^{INF}) are similar to the declarative rules: they make a recursive call to type the body of the thunk or the return expression and put the shift on top of the result.
- ($\text{ANN}^{\text{INF}}_{+}$) and ($\text{ANN}^{\text{INF}}_{-}$) are symmetric. They make a recursive call to infer the type of the annotated expression, check that the inferred type is a subtype of the annotation, and return the normalized annotation.
- (λ^{INF}) infers the type of a lambda-abstraction. It makes a recursive call to infer the type of the body in the extended context, and returns the corresponding arrow type. Notice that the algorithm also normalizes the result, which is because the annotation type P is allowed to be non-normalized.
- (Λ^{INF}) infers the type of a big lambda. Similarly to the previous case, it makes a recursive call to infer the type of the body in the extended *type* context. After that, it returns the corresponding universal type. It is also required to normalize the result, because, for instance, α^{+} might not occur in the body of the lambda, in which case the $\forall$ must be removed.
- (LET^{INF}) is defined in a standard way: it makes a recursive call to infer the type of the bound value, and then returns the type of the body in the extended context.
- ($\text{LET}^{\text{INF}}_{:@}$) is interpreted as follows. First, it infers the type of the head of the application, ensuring that it is a thunked computation $\downarrow M$; after that, it makes a recursive call to the application inference procedure, which returns the algorithmic type, whose instantiation to a declarative type must be associated with the bound variable x; then premise $T;\,\Sigma\vDash\uparrow Q\leqslant\uparrow P\dashv C_2$ together with $\Sigma\vdash C_1\,\&\,C_2=C$ check whether the instantiation to the annotated type P is possible, and if it is, the algorithm infers the type of the body in the extended context, and returns it as the result.
- ($\text{LET}^{\text{INF}}_{@}$) works similarly to ($\text{LET}^{\text{INF}}_{:@}$), However, since there is no annotation to assign the result to, the algorithm must infer the 'canonical' principal type. To do that algorithmically, we ensure that the inferred algorithmic type Q is instantiated to the minimal possible type $[\widehat{\sigma}]\,Q$. The premise Q is C-minimized by $\widehat{\sigma}$ provides the minimal instantiation of Q w.r.t. C. It guarantees that if we consider all possible substitutions satisfying the inferred constraints C, then substitution $\widehat{\sigma}$ will instantiate Q to the minimal possible type $[\widehat{\sigma}]\,Q$. This will be the principal type that we assign to the result of the application (bound to the variable x) and then we infer the type of the body in the context extended with the bound variable $x:[\widehat{\sigma}]\,Q$.
- ($\text{LET}^{\text{INF}}_{\exists}$) first, infers the existential type $\exists\overrightarrow{\alpha^{-}}.\,P$ of the value being unpacked, and since the type is guaranteed to be normalized, binds the quantified variables with $\overrightarrow{\alpha^{-}}$. Then it infers the type of the body in the appropriately extended context and checks that the inferred type does not depend on $\overrightarrow{\alpha^{-}}$ by checking well-formedness $T\vdash N$.

Finally, let us discuss the algorithmic rules of the application inference:

- ($\emptyset^{\text{INF}}_{\bullet\Rrightarrow}$) is the base case. If the list of arguments is empty, the inferred type is the type of the head, and the algorithm returns it after normalizing.

- ($\to^{\text{INF}}_{\bullet\Rrightarrow}$) is the main rule of algorithmic application inference. If the head has an arrow type $Q \to N$, we find C_1—the minimal constraint ensuring that Q is a supertype of the first argument's type. Then we make a recursive call applying N to the rest of the arguments, and merge the resulting constraint with C_1
- ($\forall^{\text{INF}}_{\bullet\Rrightarrow}$), analogously to the declarative case, is the rule ensuring the implicit elimination of the universal quantifiers. This is the place where the algorithmic variables are generated. The algorithm simply replaces the quantified variables $\overrightarrow{\alpha^+}$ with fresh algorithmic variables $\overrightarrow{\widehat{\alpha}^+}$, and makes a recursive call in the extended context.

The correctness of the algorithm consists of its soundness and completeness, which is by mutual induction in lemmas 93 and 94. The simplified result is the following.

Theorem.

$-$ $T;\Gamma \vDash c\colon N$ *implies* $T;\Gamma \vdash c\colon N$, *and* $T;\Gamma \vdash c\colon N$ *implies* $T;\Gamma \vDash c\colon \mathbf{nf}\,(N)$*;*
$+$ $T;\Gamma \vDash v\colon P$ *implies* $T;\Gamma \vdash v\colon P$, *and* $T;\Gamma \vdash v\colon P$ *implies* $T;\Gamma \vDash v\colon \mathbf{nf}\,(P)$.

4.2 Minimal Instantiation

The minimal instantiation algorithm is used to infer the type of the bound variable in the unannotated applicative let-binders, as long as there exists a principal (minimal) type. Given a positive algorithmic type P and a set of constraints C, it finds the substitution $\widehat{\sigma}$ respecting C (i.e., $\Sigma \vdash \widehat{\sigma} : C$) such that it instantiates P to the minimal type, in other words for any other substitution $\widehat{\sigma}'$ respecting C, we have $T \vdash [\widehat{\sigma}']P \geqslant [\widehat{\sigma}]P$.

The minimal instantiation algorithm is defined as follows:

Algorithm 15 (Minimal Instantiation)**.**

$\boxed{P \textit{ is } C\textit{-minimized by } \widehat{\sigma}}$

$$\frac{(\widehat{\alpha}^+ :\geqslant P) \in C}{\widehat{\alpha}^+ \textit{ is } C\textit{-minimized by } (\mathbf{nf}\,(P)/\widehat{\alpha}^+)}\quad (\textit{UVar}^{\textit{MIN}})$$

$$\frac{P \textit{ is } C\textit{-minimized by } \widehat{\sigma}}{\exists\overrightarrow{\alpha^-}.\ P \textit{ is } C\textit{-minimized by } \widehat{\sigma}}\quad (\exists^{\textit{MIN}})$$

$$\frac{\mathbf{fav}(P) \subseteq \mathbf{dom}\,(C) \qquad C|_{\mathbf{fav}(P)}\ \mathbf{singular\ with}\ \widehat{\sigma}}{P \textit{ is } C\textit{-minimized by } \widehat{\sigma}}\quad (\textit{Sing}^{\textit{MIN}})$$

4.3 Constraint Singularity

The singularity algorithm checks whether the constraint C uniquely defines the substitution satisfying it, and if it does, the algorithm returns this substitution as the result. To implement it, we define a partial function C **singular with** $\widehat{\sigma}$, taking a subtyping constraint C as an argument and returning a substitution $\widehat{\sigma}$—the only possible solution of C.

First, we define the notion of singularity on constraint entries. e **singular with** P and its negative counterpart are considered partial functions taking a constraint entry e and returning the type satisfying e if such a type is unique.

Algorithm 16 (Singular Constraint Entry)**.**

$\boxed{e\ \mathbf{singular\ with}\ P}$

$$\frac{}{\widehat{\alpha}^+ :\simeq P\ \mathbf{singular\ with\ nf}\,(P)}\quad (\simeq^{\textit{SING}}_{+})$$

$$\frac{}{\widehat{\alpha}^+ :\geqslant \exists\overrightarrow{\alpha^-}.\ \alpha^+ \ \textbf{singular with}\ \alpha^+}\ (:\geqslant\alpha^{\textit{SING}})$$

$$\frac{\textbf{nf}\ (N) = \alpha^-_i}{\widehat{\alpha}^+ :\geqslant \exists\overrightarrow{\alpha^-}.\ {\downarrow}N \ \textbf{singular with}\ \exists\alpha^-.\ {\downarrow}\alpha^-}\ (:\geqslant{\downarrow}^{\textit{SING}})$$

$\boxed{e \ \textbf{singular with}\ N}$

$$\frac{}{\widehat{\alpha}^- :\simeq N \ \textbf{singular with nf}\ (N)}\ (\simeq^{\textit{SING}}_-)$$

- $(\simeq^{\text{SING}}_-)$ and $(\simeq^{\text{SING}}_+)$ are symmetric. If the constraint entry says that a variable must be equivalent to a type T, then it is evidently singular, and the only (up-to-equivalence) type instantiating this variable could be T. This way, we return its normal form.
- $(:\geqslant\alpha^{\text{SING}})$ implies that the only (normalized) solution of $\widehat{\alpha}^+ :\geqslant \exists\overrightarrow{\alpha^-}.\ \alpha^+$ is α^+ (it will be shown in lemma 17).
- $(:\geqslant{\downarrow}^{\text{SING}})$ is perhaps the least obvious rule. Having a type $\exists\overrightarrow{\alpha^-}.\ {\downarrow}N$, suppose that N is not equivalent to any just bound variable $\alpha^-{}_i \in \overrightarrow{\alpha^-}$. Then the type $\exists\overrightarrow{\alpha^-}.\ {\downarrow}N$ has a proper supertype: $\exists\overrightarrow{\alpha^-}.\ {\downarrow}\alpha^-{}_1$, and thus the constraint is not singular. Otherwise, if N is equivalent to some $\alpha^-{}_i$, any supertype of $\exists\overrightarrow{\alpha^-}.\ {\downarrow}\alpha^-{}_i$ is equivalent to it, and thus, the constraint has a unique solution.

Next, we extrapolate the singularity function on constraints—sets of constraint entries. We require C to be a set of singular constraints, and the resulting substitution sends each variable from **dom** (C) to the unique type satisfying the corresponding constraint.

Algorithm 17. *C **singular with** $\widehat{\sigma}$ means that*

*(1) for any positive $e \in C$, there exists P such that e **singular with** P, and for any negative $e \in C$, there exists N such that e **singular with** N;*

(2) $\widehat{\sigma}$ is defined as follows:

$$[\widehat{\sigma}]\widehat{\beta}^+ = \begin{cases} P & \textit{if there is } e \in \textbf{dom}\ (C) \textit{ restricting } \widehat{\beta}^+ \textit{ and } e \ \textbf{singular with}\ P \\ \widehat{\beta}^+ & \textit{otherwise} \end{cases}$$

$$[\widehat{\sigma}]\widehat{\beta}^- = \begin{cases} N & \textit{if there is } e \in \textbf{dom}\ (C) \textit{ restricting } \widehat{\beta}^- \textit{ and } e \ \textbf{singular with}\ N \\ \widehat{\beta}^- & \textit{otherwise} \end{cases}$$

The correctness of the singularity algorithm is formulated as follows:

Theorem. *Suppose that C is a subtyping constraint. Then C **singular with** $\widehat{\sigma}$ holds if and only if $\widehat{\sigma}$ is the only (up-to-equivalence on **dom** (C)) normalized substitution satisfying C.*

5 THEOREM STATEMENTS: DECLARATIVE

5.1 Type Well-Formedness

Lemma 1 (Soundness of type well-formedness).

+ *If* $T \vdash P$ *then* $\mathbf{fv}\,(P) \subseteq T$,
− *if* $T \vdash N$ *then* $\mathbf{fv}\,(N) \subseteq T$.

Lemma 2 (Completeness of type well-formedness). *In the well-formedness judgment, only used variables matter:*

+ *if* $T_1 \cap \mathbf{fv}\,P = T_2 \cap \mathbf{fv}\,P$ *then* $T_1 \vdash P \iff T_2 \vdash P$,
− *if* $T_1 \cap \mathbf{fv}\,N = T_2 \cap \mathbf{fv}\,N$ *then* $T_1 \vdash N \iff T_2 \vdash N$.

Corollary 1 (Context Strengthening).

+ *If* $T \vdash P$ *then* $\mathbf{fv}\,(P) \vdash P$;
− *If* $T \vdash N$ *then* $\mathbf{fv}\,(N) \vdash N$.

Corollary 2 (Well-formedness Context Weakening). *Suppose that* $T_1 \subseteq T_2$, *then*

+ *if* $T_1 \vdash P$ *then* $T_2 \vdash P$,
− *if* $T_1 \vdash N$ *then* $T_2 \vdash N$.

Lemma 3 (Well-formedness agrees with substitution). *Suppose that* $T_2 \vdash \sigma : T_1$. *Then*

+ $T, T_1 \vdash P$ *implies* $T, T_2 \vdash [\sigma]P$, *and*
− $T, T_1 \vdash N$ *implies* $T, T_2 \vdash [\sigma]N$.

5.2 Substitution

Lemma 4 (Substitution strengthening). *Restricting the substitution to the free variables of the substitution subject does not affect the result. Suppose that* σ *is a substitution,* P *and* N *are types. Then*

+ $[\sigma]P = [\sigma|_{\mathbf{fv}\,P}]P$,
− $[\sigma]N = [\sigma|_{\mathbf{fv}\,N}]N$

Lemma 5 (Signature of a restricted substitution). *If* $T_2 \vdash \sigma : T_1$ *then* $T_2 \vdash \sigma|_{vars} : T_1 \cap vars$.

Lemma 6. *Suppose that* σ *is a substitution with signature* $T_2 \vdash \sigma : T_1$. *Then if* $vars$ *is disjoint from* T_1, *then* $\sigma|_{vars} = \mathbf{id}$.

Corollary 3 (Application of a disjoint substitution). *Suppose that* σ *is a substitution with signature* $T_2 \vdash \sigma : T_1$. *Then*

+ *if* $T_1 \cap \mathbf{fv}\,(Q) = \emptyset$ *then* $[\sigma]Q = Q$;
− *if* $T_1 \cap \mathbf{fv}\,(N) = \emptyset$ *then* $[\sigma]N = N$.

Lemma 7 (Substitution range weakening). *Suppose that* $T_2 \subseteq T_2'$ *are contexts and* σ *is a substitution. Then* $T_2 \vdash \sigma : T_1$ *implies* $T_2' \vdash \sigma : T_1$.

Lemma 8 (Subsitutions Equivalent on Free Variables). *Suppose that* $T' \subseteq T$, σ_1 *and* σ_2 *are substitutions of signature* $T \vdash \sigma_i : T'$. *Then*

+ *for a type* $T \vdash P$, *if* $T \vdash [\sigma_1]P \simeq^{\leqslant} [\sigma_2]P$ *then* $T \vdash \sigma_1 \simeq^{\leqslant} \sigma_2 : \mathbf{fv}\,P \cap T'$;
− *for a type* $T \vdash N$, *if* $T \vdash [\sigma_1]N \simeq^{\leqslant} [\sigma_2]N$ *then* $T \vdash \sigma_1 \simeq^{\leqslant} \sigma_2 : \mathbf{fv}\,N \cap T'$.

Lemma 9 (Substitution composition well-formedness). *If $T_1' \vdash \sigma_1 : T_1$ and $T_2' \vdash \sigma_2 : T_2$, then $T_1', T_2' \vdash \sigma_2 \circ \sigma_1 : T_1, T_2$.*

Lemma 10 (Substitution monadic composition well-formedness). *If $T_1' \vdash \sigma_1 : T_1$ and $T_2' \vdash \sigma_2 : T_2$, then $T_2' \vdash \sigma_2 \lll \sigma_1 : T_1$.*

Lemma 11 (Substitution composition). *If $T_1' \vdash \sigma_1 : T_1$, $T_2' \vdash \sigma_2 : T_2$, $T_1 \cap T_2' = \emptyset$ and $T_1 \cap T_2 = \emptyset$ then $\sigma_2 \circ \sigma_1 = (\sigma_2 \lll \sigma_1) \circ \sigma_2$.*

Corollary 4 (Substitution composition commutativity). *If $T_1' \vdash \sigma_1 : T_1$, $T_2' \vdash \sigma_2 : T_2$, and $T_1 \cap T_2 = \emptyset$, $T_1 \cap T_2' = \emptyset$, and $T_1' \cap T_2 = \emptyset$ then $\sigma_2 \circ \sigma_1 = \sigma_1 \circ \sigma_2$.*

Lemma 12 (Substitution domain weakening). *If $T_2 \vdash \sigma : T_1$ then $T_2, T' \vdash \sigma : T_1, T'$*

Lemma 13 (Free variables after substitution). *Suppose that $T_2 \vdash \sigma : T_1$, then*

+ *for a type P, the free variables of $[\sigma]P$ are bounded in the following way: $\mathbf{fv}(P) \setminus T_1 \subseteq \mathbf{fv}([\sigma]P) \subseteq (\mathbf{fv}(P) \setminus T_1) \cup T_2$;*
− *for a type N, the free variables of $[\sigma]P$ are bounded in the following way: $\mathbf{fv}(N) \setminus T_1 \subseteq \mathbf{fv}([\sigma]N) \subseteq (\mathbf{fv}(N) \setminus T_1) \cup T_2$.*

Lemma 14 (Free variables of a variable image). *Suppose that σ is an arbitrary substitution, Then*

+ *if $\alpha^{\pm} \in \mathbf{fv}(P)$ then $\mathbf{fv}([\sigma]\alpha^{\pm}) \subseteq \mathbf{fv}([\sigma]P)$,*
− *if $\alpha^{\pm} \in \mathbf{fv}(N)$ then $\mathbf{fv}([\sigma]\alpha^{\pm}) \subseteq \mathbf{fv}([\sigma]N)$.*

5.3 Declarative Subtyping

Lemma 15 (Free Variable Propagation). *In the judgments of negative subtyping or positive supertyping, free variables propagate left to right. For a context T,*

− *if $T \vdash N \leqslant M$ then $\mathbf{fv}(N) \subseteq \mathbf{fv}(M)$*
+ *if $T \vdash P \geqslant Q$ then $\mathbf{fv}(P) \subseteq \mathbf{fv}(Q)$*

Corollary 5 (Free Variables of mutual subtypes).

− *If $T \vdash N \simeq^{\leqslant} M$ then $\mathbf{fv}\, N = \mathbf{fv}\, M$,*
+ *If $T \vdash P \simeq^{\leqslant} Q$ then $\mathbf{fv}\, P = \mathbf{fv}\, Q$*

Corollary 6. *Suppose that all the types below are well-formed in T and $T' \subseteq T$. Then*

+ *$T \vdash P \simeq^{\leqslant} Q$ implies $T' \vdash P \iff T' \vdash Q$*
− *$T \vdash N \simeq^{\leqslant} M$ implies $T' \vdash N \iff T' \vdash M$*

Lemma 16 (Decomposition of quantifier rules). *Assuming that $\overrightarrow{\alpha^+}$, $\overrightarrow{\beta^+}$, $\overrightarrow{\alpha^-}$, and $\overrightarrow{\alpha^-}$ are disjoint from T,*

$-_R$ *$T \vdash N \leqslant \forall\overrightarrow{\beta^+}.\, M$ holds if and only if $T, \overrightarrow{\beta^+} \vdash N \leqslant M$;*
$+_R$ *$T \vdash P \geqslant \exists\overrightarrow{\beta^-}.\, Q$ holds if and only if $T, \overrightarrow{\beta^-} \vdash P \geqslant Q$;*
$-_L$ *suppose $M \neq \forall\ldots$ then $T \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant M$ holds if and only if $T \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N \leqslant M$ for some $T \vdash \overrightarrow{P}$;*
$+_L$ *suppose $Q \neq \exists\ldots$ then $T \vdash \exists\overrightarrow{\alpha^-}.\, P \geqslant Q$ holds if and only if $T \vdash [\overrightarrow{N}/\overrightarrow{\alpha^-}]P \geqslant Q$ for some $T \vdash \overrightarrow{N}$.*

Corollary 7 (Redundant quantifier elimination).

$-_L$ *Suppose that* $\overrightarrow{\alpha^+} \cap \mathbf{fv}\,(N) = \emptyset$ *then* $T \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant M$ *holds if and only if* $T \vdash N \leqslant M$*;*
$-_R$ *Suppose that* $\overrightarrow{\alpha^+} \cap \mathbf{fv}\,(M) = \emptyset$ *then* $T \vdash N \leqslant \forall\overrightarrow{\alpha^+}.\, M$ *holds if and only if* $T \vdash N \leqslant M$*;*
$+_L$ *Suppose that* $\overrightarrow{\alpha^-} \cap \mathbf{fv}\,(P) = \emptyset$ *then* $T \vdash \exists\overrightarrow{\alpha^-}.\, P \geqslant Q$ *holds if and only if* $T \vdash P \geqslant Q$*.*
$+_R$ *Suppose that* $\overrightarrow{\alpha^-} \cap \mathbf{fv}\,(Q) = \emptyset$ *then* $T \vdash P \geqslant \exists\overrightarrow{\alpha^-}.\, Q$ *holds if and only if* $T \vdash P \geqslant Q$*.*

Lemma 17 (Subtypes and supertypes of a variable). *Assuming* $T \vdash \alpha^-$, $T \vdash \alpha^+$, $T \vdash N$, *and* $T \vdash P$,

$+$ *if* $T \vdash P \geqslant \exists\overrightarrow{\alpha^-}.\, \alpha^+$ *or* $T \vdash \exists\overrightarrow{\alpha^-}.\, \alpha^+ \geqslant P$ *then* $P = \exists\overrightarrow{\beta^-}.\, \alpha^+$ *(for some potentially empty* $\overrightarrow{\beta^-}$*)*
$-$ *if* $T \vdash N \leqslant \forall\overrightarrow{\alpha^+}.\, \alpha^-$ *or* $T \vdash \forall\overrightarrow{\alpha^+}.\, \alpha^- \leqslant N$ *then* $N = \forall\overrightarrow{\beta^+}.\, \alpha^-$ *(for some potentially empty* $\overrightarrow{\beta^+}$*)*

Corollary 8 (Variables have no proper subtypes and supertypes). *Assuming that all mentioned types are well-formed in* T,

$$\begin{aligned}
T \vdash P \geqslant \alpha^+ \quad&\Longleftrightarrow\quad P = \exists\overrightarrow{\beta^-}.\, \alpha^+ \quad\Longleftrightarrow\quad T \vdash P \simeq^{\leqslant} \alpha^+ \quad\Longleftrightarrow\quad P \simeq^{D} \alpha^+\\
T \vdash \alpha^+ \geqslant P \quad&\Longleftrightarrow\quad P = \exists\overrightarrow{\beta^-}.\, \alpha^+ \quad\Longleftrightarrow\quad T \vdash P \simeq^{\leqslant} \alpha^+ \quad\Longleftrightarrow\quad P \simeq^{D} \alpha^+\\
T \vdash N \leqslant \alpha^- \quad&\Longleftrightarrow\quad N = \forall\overrightarrow{\beta^+}.\, \alpha^- \quad\Longleftrightarrow\quad T \vdash N \simeq^{\leqslant} \alpha^- \quad\Longleftrightarrow\quad N \simeq^{D} \alpha^-\\
T \vdash \alpha^- \leqslant N \quad&\Longleftrightarrow\quad N = \forall\overrightarrow{\beta^+}.\, \alpha^- \quad\Longleftrightarrow\quad T \vdash N \simeq^{\leqslant} \alpha^- \quad\Longleftrightarrow\quad N \simeq^{D} \alpha^-
\end{aligned}$$

Lemma 18 (Subtyping context irrelevance). *Suppose that all the mentioned types are well-formed in* T_1 *and* T_2*. Then*

$+$ $T_1 \vdash P \geqslant Q$ *is equivalent to* $T_2 \vdash P \geqslant Q$*;*
$-$ $T_1 \vdash N \leqslant M$ *is equivalent to* $T_2 \vdash N \leqslant M$*.*

Lemma 19 (Weakening of subtyping context). *Suppose* T_1 *and* T_2 *are contexts and* $T_1 \subseteq T_2$*. Then*

$+$ $T_1 \vdash P \geqslant Q$ *implies* $T_2 \vdash P \geqslant Q$*;*
$-$ $T_1 \vdash N \leqslant M$ *implies* $T_2 \vdash N \leqslant M$*.*

Lemma 20 (Reflexivity of subtyping). *Assuming all the types are well-formed in* T,

$-$ $T \vdash N \leqslant N$
$+$ $T \vdash P \geqslant P$

Lemma 21 (Substitution preserves subtyipng). *Suppose that all mentioned types are well-formed in* T_1*, and* σ *is a substitution* $T_2 \vdash \sigma : T_1$*.*

$-$ *If* $T_1 \vdash N \leqslant M$ *then* $T_2 \vdash [\sigma]N \leqslant [\sigma]M$*.*
$+$ *If* $T_1 \vdash P \geqslant Q$ *then* $T_2 \vdash [\sigma]P \geqslant [\sigma]Q$*.*

Corollary 9 (Substitution preserves subtyping induced equivalence). *Suppose that* $T \vdash \sigma : T_1$*. Then*

$+$ *if* $T_1 \vdash P$, $T_1 \vdash Q$*, and* $T_1 \vdash P \simeq^{\leqslant} Q$ *then* $T \vdash [\sigma]P \simeq^{\leqslant} [\sigma]Q$
$-$ *if* $T_1 \vdash N$, $T_1 \vdash M$*, and* $T_1 \vdash N \simeq^{\leqslant} M$ *then* $T \vdash [\sigma]N \simeq^{\leqslant} [\sigma]M$

Lemma 22 (Transitivity of subtyping). *Assuming the types are well-formed in* T,

$-$ *if* $T \vdash N_1 \leqslant N_2$ *and* $T \vdash N_2 \leqslant N_3$ *then* $T \vdash N_1 \leqslant N_3$,
$+$ *if* $T \vdash P_1 \geqslant P_2$ *and* $T \vdash P_2 \geqslant P_3$ *then* $T \vdash P_1 \geqslant P_3$*.*

Corollary 10 (Transitivity of equivalence). *Assuming the types are well-formed in* T,

$-$ *if* $T \vdash N_1 \simeq^{\leqslant} N_2$ *and* $T \vdash N_2 \simeq^{\leqslant} N_3$ *then* $T \vdash N_1 \simeq^{\leqslant} N_3$,
$+$ *if* $T \vdash P_1 \simeq^{\leqslant} P_2$ *and* $T \vdash P_2 \simeq^{\leqslant} P_3$ *then* $T \vdash P_1 \simeq^{\leqslant} P_3$*.*

5.4 Equivalence

Lemma 23 (Declarative Equivalence is invariant under bijections). *Suppose μ is a bijection $\mu : vars_1 \leftrightarrow vars_2$, then*

+ *$P_1 \simeq^D P_2$ implies $[\mu]P_1 \simeq^D [\mu]P_2$, and there exists an inference tree of $[\mu]P_1 \simeq^D [\mu]P_2$ with the same shape as the one inferring $P_1 \simeq^D P_2$;*
− *$N_1 \simeq^D N_2$ implies $[\mu]N_1 \simeq^D [\mu]N_2$, and there exists an inference tree of $[\mu]N_1 \simeq^D [\mu]N_2$ with the same shape as the one inferring $N_1 \simeq^D N_2$.*

Lemma 24. *The set of free variables is invariant under equivalence.*

− *If $N \simeq^D M$ then $\mathbf{fv}\, N = \mathbf{fv}\, M$ (as sets)*
+ *If $P \simeq^D Q$ then $\mathbf{fv}\, P = \mathbf{fv}\, Q$ (as sets)*

Lemma 25 (Declarative equivalence is transitive).

+ *if $P_1 \simeq^D P_2$ and $P_2 \simeq^D P_3$ then $P_1 \simeq^D P_3$,*
− *if $N_1 \simeq^D N_2$ and $N_2 \simeq^D N_3$ then $N_1 \simeq^D N_3$.*

Lemma 26 (Type well-formedness is invariant under equivalence). *Mutual subtyping implies declarative equivalence.*

+ *if $P \simeq^D Q$ then $T \vdash P \iff T \vdash Q$,*
− *if $N \simeq^D M$ then $T \vdash N \iff T \vdash M$*

Lemma 27 (Soundness of equivalence). *Declarative equivalence implies mutual subtyping.*

+ *if $T \vdash P$, $T \vdash Q$, and $P \simeq^D Q$ then $T \vdash P \simeq^{\leqslant} Q$,*
− *if $T \vdash N$, $T \vdash M$, and $N \simeq^D M$ then $T \vdash N \simeq^{\leqslant} M$.*

Lemma 28 (Subtyping induced by disjoint substitutions). *Suppose that $T \vdash \sigma_1 : T_1$ and $T \vdash \sigma_2 : T_1$, where $T_i \subseteq T$ and $T_1 \cap T_2 = \emptyset$. Then*

− *assuming $T \vdash N$, $T \vdash [\sigma_1]N \leqslant [\sigma_2]N$ implies $T \vdash \sigma_i \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, N$*
+ *assuming $T \vdash P$, $T \vdash [\sigma_1]P \geqslant [\sigma_2]P$ implies $T \vdash \sigma_i \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, P$*

Corollary 11 (Substitution cannot induce proper subtypes or supertypes). *Assuming all mentioned types are well-formed in T and σ is a substitution $T \vdash \sigma : T$,*

$$\begin{aligned}
T \vdash [\sigma]N \leqslant N &\Rightarrow T \vdash [\sigma]N \simeq^{\leqslant} N \text{ and } T \vdash \sigma \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, N\\
T \vdash N \leqslant [\sigma]N &\Rightarrow T \vdash N \simeq^{\leqslant} [\sigma]N \text{ and } T \vdash \sigma \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, N\\
T \vdash [\sigma]P \geqslant P &\Rightarrow T \vdash [\sigma]P \simeq^{\leqslant} P \text{ and } T \vdash \sigma \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, P\\
T \vdash P \geqslant [\sigma]P &\Rightarrow T \vdash P \simeq^{\leqslant} [\sigma]P \text{ and } T \vdash \sigma \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, P
\end{aligned}$$

Lemma 29 (Mutual substitution and subtyping). *Assuming that the mentioned types (P, Q, N, and M) are well-formed in T and that the substitutions (σ_1 and σ_2) have signature $T \vdash \sigma_i : T$,*

+ *if $T \vdash [\sigma_1]P \geqslant Q$ and $T \vdash [\sigma_2]Q \geqslant P$ then there exists a bijection $\mu : \mathbf{fv}\, P \leftrightarrow \mathbf{fv}\, Q$ such that $T \vdash \sigma_1 \simeq^{\leqslant} \mu : \mathbf{fv}\, P$ and $T \vdash \sigma_2 \simeq^{\leqslant} \mu^{-1} : \mathbf{fv}\, Q$;*
− *if $T \vdash [\sigma_1]N \leqslant M$ and $T \vdash [\sigma_2]N \leqslant M$ then there exists a bijection $\mu : \mathbf{fv}\, N \leftrightarrow \mathbf{fv}\, M$ such that $T \vdash \sigma_1 \simeq^{\leqslant} \mu : \mathbf{fv}\, N$ and $T \vdash \sigma_2 \simeq^{\leqslant} \mu^{-1} : \mathbf{fv}\, M$.*

Lemma 30 (Equivalent substitution act equivalently). *Suppose that $T' \vdash \sigma_1 : T$ and $T' \vdash \sigma_2 : T$ are substitutions equivalent on their domain, that is $T' \vdash \sigma_1 \simeq^{\leqslant} \sigma_2 : T$. Then*

+ *for any $T \vdash P$, $T' \vdash [\sigma_1]P \simeq^{\leqslant} [\sigma_2]P$;*
− *for any $T \vdash N$, $T' \vdash [\sigma_1]N \simeq^{\leqslant} [\sigma_2]N$.*

Lemma 31 (Equivalence of polymorphic types).

− *For $T \vdash \forall\overrightarrow{\alpha^+}.\, N$ and $T \vdash \forall\overrightarrow{\beta^+}.\, M$,*
if $T \vdash \forall\overrightarrow{\alpha^+}.\, N \simeq^{\leqslant} \forall\overrightarrow{\beta^+}.\, M$ then there exists a bijection $\mu : \overrightarrow{\beta^+} \cap \mathbf{fv}\, M \leftrightarrow \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N$ such that $T, \overrightarrow{\alpha^+} \vdash N \simeq^{\leqslant} [\mu]M$,
+ *For $T \vdash \exists\overrightarrow{\alpha^-}.\, P$ and $T \vdash \exists\overrightarrow{\beta^-}.\, Q$,*
if $T \vdash \exists\overrightarrow{\alpha^-}.\, P \simeq^{\leqslant} \exists\overrightarrow{\beta^-}.\, Q$ then there exists a bijection $\mu : \overrightarrow{\beta^-} \cap \mathbf{fv}\, Q \leftrightarrow \overrightarrow{\alpha^-} \cap \mathbf{fv}\, P$ such that $T, \overrightarrow{\beta^-} \vdash P \simeq^{\leqslant} [\mu]Q$.

Lemma 32 (Completeness of Equivalence). *Mutual subtyping implies declarative equivalence. Assuming all the types below are well-formed in T:*

+ *if $T \vdash P \simeq^{\leqslant} Q$ then $P \simeq^{D} Q$,*
− *if $T \vdash N \simeq^{\leqslant} M$ then $N \simeq^{D} M$.*

5.5 Variable Ordering

Observation 1 (Ordering is deterministic). *If* **ord** *vars* **in** $N = \vec{\alpha}_1$ *and* **ord** *vars* **in** $N = \vec{\alpha}_2$ *then* $\vec{\alpha}_1 = \vec{\alpha}_2$*. If* **ord** *vars* **in** $P = \vec{\alpha}_1$ *and* **ord** *vars* **in** $P = \vec{\alpha}_2$ *then* $\vec{\alpha}_1 = \vec{\alpha}_2$*. This way, we can use* **ord** *vars* **in** N *and as a function on N, and* **ord** *vars* **in** P *as a function on P.*

Lemma 33 (Soundness of variable ordering). *Variable ordering extracts used free variables.*

− **ord** *vars* **in** N = *vars* $\cap$ **fv** N *(as sets)*
+ **ord** *vars* **in** P = *vars* $\cap$ **fv** P *(as sets)*

Corollary 12 (Additivity of ordering). *Variable ordering is additive (in terms of set union) with respect to its first argument.*

− **ord** $(vars_1 \cup vars_2)$ **in** N = **ord** $vars_1$ **in** N $\cup$ **ord** $vars_2$ **in** N *(as sets)*
+ **ord** $(vars_1 \cup vars_2)$ **in** P = **ord** $vars_1$ **in** P $\cup$ **ord** $vars_2$ **in** P *(as sets)*

Lemma 34 (Weakening of ordering). *Only used variables matter in the first argument of the ordering,*

− **ord** $(vars \cap \mathbf{fv}\, N)$ **in** N = **ord** *vars* **in** N
+ **ord** $(vars \cap \mathbf{fv}\, P)$ **in** P = **ord** *vars* **in** P

Corollary 13 (Idempotency of ordering).

− *If* **ord** *vars* **in** $N = \vec{\alpha}$ *then* **ord** $\vec{\alpha}$ **in** $N = \vec{\alpha}$,
+ *If* **ord** *vars* **in** $P = \vec{\alpha}$ *then* **ord** $\vec{\alpha}$ **in** $P = \vec{\alpha}$;

Lemma 35 (Distributivity of renaming over variable ordering). *Suppose that μ is a bijection between two sets of variables $\mu : A \leftrightarrow B$.*

− *If μ is collision-free on vars and* **fv** N *then* $[\mu]($**ord** *vars* **in** $N)$ = **ord** $([\mu]\,vars)$ **in** $[\mu]N$
+ *If μ is collision-free on vars and* **fv** P *then* $[\mu]($**ord** *vars* **in** $P)$ = **ord** $([\mu]\,vars)$ **in** $[\mu]P$

Lemma 36 (Ordering is not affected by independent substitutions). *Suppose that $T_2 \vdash \sigma : T_1$, i.e. σ maps variables from T_1 into types taking free variables from T_2, and vars is a set of variables disjoint with both T_1 and T_2, N and P are types. Then*

- $-$ $\mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]N = \mathbf{ord}\ vars\ \mathbf{in}\ N$
- $+$ $\mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]P = \mathbf{ord}\ vars\ \mathbf{in}\ P$

Lemma 37 (Completeness of variable ordering). *Variable ordering is invariant under equivalence. For arbitrary vars,*

- $-$ *If* $N \simeq^D M$ *then* $\mathbf{ord}\ vars\ \mathbf{in}\ N = \mathbf{ord}\ vars\ \mathbf{in}\ M$ *(as lists)*
- $+$ *If* $P \simeq^D Q$ *then* $\mathbf{ord}\ vars\ \mathbf{in}\ P = \mathbf{ord}\ vars\ \mathbf{in}\ Q$ *(as lists)*

5.6 Normaliztaion

Observation 2 (Normalization is deterministic). *If* $\mathbf{nf}\ (N) = M$ *and* $\mathbf{nf}\ (N) = M'$ *then* $M = M'$. *If* $\mathbf{nf}\ (P) = Q$ *and* $\mathbf{nf}\ (P) = Q'$ *then* $Q = Q'$. *This way, we can use normalization as a function.*

Lemma 38. *Free variables are not changed by the normalization*

- $-$ $\mathbf{fv}\ N = \mathbf{fv}\ \mathbf{nf}\ (N)$
- $+$ $\mathbf{fv}\ P = \mathbf{fv}\ \mathbf{nf}\ (P)$

Lemma 39 (Soundness of normalization).

- $-$ $N \simeq^D \mathbf{nf}\ (N)$
- $+$ $P \simeq^D \mathbf{nf}\ (P)$

Corollary 14 (Normalization preserves well-formedness).

- $+$ $T \vdash P \iff T \vdash \mathbf{nf}\ (P)$,
- $-$ $T \vdash N \iff T \vdash \mathbf{nf}\ (N)$

Corollary 15 (Normalization preserves well-formedness of substitution). $T_2 \vdash \sigma : T_1 \iff T_2 \vdash \mathbf{nf}\ (\sigma) : T_1$

Lemma 40 (Normalization preserves substitution signature). *Suppose that σ is a substitution, T_1 and T_2 are contexts. Then $T_2 \vdash \sigma : T_1$ implies $T_2 \vdash \mathbf{nf}\ (\sigma) : T_1$.*

Corollary 16 (Normalization is sound w.r.t. subtyping-induced equivalence).

- $+$ *if* $T \vdash P$ *then* $T \vdash P \simeq^{\leqslant} \mathbf{nf}\ (P)$,
- $-$ *if* $T \vdash N$ *then* $T \vdash N \simeq^{\leqslant} \mathbf{nf}\ (N)$.

Corollary 17 (Normalization preserves subtyping). *Assuming all the types are well-formed in context T,*

- $+$ $T \vdash P \geqslant Q \iff T \vdash \mathbf{nf}\ (P) \geqslant \mathbf{nf}\ (Q)$,
- $-$ $T \vdash N \leqslant M \iff T \vdash \mathbf{nf}\ (N) \leqslant \mathbf{nf}\ (M)$.

Corollary 18 (Normalization preserves ordering). *For any vars,*

- $-$ $\mathbf{ord}\ vars\ \mathbf{in}\ \mathbf{nf}\ (N) = \mathbf{ord}\ vars\ \mathbf{in}\ M$
- $+$ $\mathbf{ord}\ vars\ \mathbf{in}\ \mathbf{nf}\ (P) = \mathbf{ord}\ vars\ \mathbf{in}\ Q$

Lemma 41 (Distributivity of normalization over substitution). *Normalization of a term distributes over substitution. Suppose that σ is a substitution, N and P are types. Then*

- $-$ $\mathbf{nf}([\sigma]N) = [\mathbf{nf}(\sigma)]\mathbf{nf}(N)$
- $+$ $\mathbf{nf}([\sigma]P) = [\mathbf{nf}(\sigma)]\mathbf{nf}(P)$

where $\mathbf{nf}(\sigma)$ *means pointwise normalization:* $[\mathbf{nf}(\sigma)]\alpha^- = \mathbf{nf}([\sigma]\alpha^-)$.

Corollary 19 (Commutativity of normalization and renaming). *Normalization of a term commutes with renaming. Suppose that μ is a bijection between two sets of variables $\mu : A \leftrightarrow B$. Then*

- $-$ $\mathbf{nf}([\mu]N) = [\mu]\mathbf{nf}(N)$
- $+$ $\mathbf{nf}([\mu]P) = [\mu]\mathbf{nf}(P)$

Lemma 42 (Completeness of Normalization w.r.t. Declarative Equivalence). *Normalization returns the same representative for equivalent types.*

- $-$ *If* $N \simeq^D M$ *then* $\mathbf{nf}(N) = \mathbf{nf}(M)$,
- $+$ *if* $P \simeq^D Q$ *then* $\mathbf{nf}(P) = \mathbf{nf}(Q)$.

Lemma 43 (Algorithmization of Declarative Equivalence). *Declarative equivalence is the equality of normal forms.*

- $+$ $P \simeq^D Q \iff \mathbf{nf}(P) = \mathbf{nf}(Q)$,
- $-$ $N \simeq^D M \iff \mathbf{nf}(N) = \mathbf{nf}(M)$.

Corollary 20 (Completeness of Normalization w.r.t. Subtyping-Induced Equivalence). *Assuming all the types below are well-formed in T:*

- $+$ *if* $T \vdash P \simeq^\leqslant Q$ *then* $\mathbf{nf}(P) = \mathbf{nf}(Q)$,
- $-$ *if* $T \vdash N \simeq^\leqslant M$ *then* $\mathbf{nf}(N) = \mathbf{nf}(M)$.

Lemma 44 (Idempotence of normalization). *Normalization is idempotent*

- $-$ $\mathbf{nf}(\mathbf{nf}(N)) = \mathbf{nf}(N)$
- $+$ $\mathbf{nf}(\mathbf{nf}(P)) = \mathbf{nf}(P)$

Lemma 45. *The result of a substitution is normalized if and only if the initial type and the substitution are normalized.*

Suppose that σ is a substitution $T_2 \vdash \sigma : T_1$, P is a positive type ($T_1 \vdash P$), N is a negative type ($T_1 \vdash N$). Then

- $+$ $[\sigma]P$ *is normal* $\iff \begin{cases} \sigma|_{\mathbf{fv}(P)} & \text{is normal} \\ P & \text{is normal} \end{cases}$
- $-$ $[\sigma]N$ *is normal* $\iff \begin{cases} \sigma|_{\mathbf{fv}(N)} & \text{is normal} \\ N & \text{is normal} \end{cases}$

Lemma 46 (Algorithmization of subtyping-induced equivalence). *Mutual subtyping is the equality of normal forms. Assuming all the types below are well-formed in T:*

- $+$ $T \vdash P \simeq^\leqslant Q \iff \mathbf{nf}(P) = \mathbf{nf}(Q)$,
- $-$ $T \vdash N \simeq^\leqslant M \iff \mathbf{nf}(N) = \mathbf{nf}(M)$.

Corollary 21 (Substitution preserves declarative equivalence). *Suppose that σ is a substitution. Then*

- $+$ $P \simeq^D Q$ *implies* $[\sigma]P \simeq^D [\sigma]Q$
- $-$ $N \simeq^D M$ *implies* $[\sigma]N \simeq^D [\sigma]M$

6 DECLARATIVE TYPING

Lemma 47. *If* $T\,;\Gamma \vdash N_1 \bullet \vec{v} \Rrightarrow M$ *and* $T \vdash N_1 \simeq^{\leqslant} N_2$ *then* $T\,;\Gamma \vdash N_2 \bullet \vec{v} \Rrightarrow M$.

Lemma 48 (Declarative typing is preserved under context equivalence). *Assuming* $T \vdash \Gamma_1$, $T \vdash \Gamma_2$, *and* $T \vdash \Gamma_1 \simeq^{\leqslant} \Gamma_2$*:*

+ *for any tree* T_1 *inferring* $T;\Gamma_1 \vdash v\colon P$, *there exists a tree* T_2 *inferring* $T;\Gamma_2 \vdash v\colon P$.
− *for any tree* T_1 *inferring* $T;\Gamma_1 \vdash c\colon N$, *there exists a tree* T_2 *inferring* $T;\Gamma_2 \vdash c\colon N$.
• *for any tree* T_1 *inferring* $T;\Gamma_1 \vdash N \bullet \vec{v} \Rrightarrow M$, *there exists a tree* T_2 *inferring* $T;\Gamma_2 \vdash N \bullet \vec{v} \Rrightarrow M$.

7 THEOREM STATEMENTS: ALGORITHMIC

7.1 Algorithmic Type Well-formedness

Lemma 49. *If* $T\,;\Gamma \vdash N_1 \bullet \vec{v} \Rrightarrow M$ *and* $T \vdash N_1 \simeq^{\leqslant} N_2$ *then* $T\,;\Gamma \vdash N_2 \bullet \vec{v} \Rrightarrow M$.

Lemma 50 (Soundness of algorithmic type well-formedness).

+ *if* $T\,;\Upsilon \vdash P$ *then* $\mathbf{fv}\,(P) \subseteq T$ *and* $\mathbf{fav}(P) \subseteq \Upsilon$;
− *if* $T\,;\Upsilon \vdash N$ *then* $\mathbf{fv}\,(N) \subseteq T$ *and* $\mathbf{fav}(N) \subseteq \Upsilon$.

Lemma 51 (Completeness of algorithmic type well-formedness). *In the well-formedness judgment, only used variables matter:*

+ *if* $T_1 \cap \mathbf{fv}\,P = T_2 \cap \mathbf{fv}\,P$ *and* $\Upsilon_1 \cap \mathbf{fav} P = \Upsilon_2 \cap \mathbf{fav} P$ *then* $T_1\,;\Upsilon_1 \vdash P \iff T_2\,;\Upsilon_2 \vdash P$, *and*
− *if* $T_1 \cap \mathbf{fv}\,N = T_2 \cap \mathbf{fv}\,N$ *and* $\Upsilon_1 \cap \mathbf{fav} N = \Upsilon_2 \cap \mathbf{fav} N$ *then* $T_1\,;\Upsilon_1 \vdash N \iff T_2\,;\Upsilon_2 \vdash N$.

Lemma 52 (Variable algorithmization agrees with well-formedness).

+ $T, \overrightarrow{\alpha^-} \vdash P$ *implies* $T\,;\overrightarrow{\widehat{\alpha}^-} \vdash [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]P$;
− $T, \overrightarrow{\alpha^-} \vdash N$ *implies* $T\,;\overrightarrow{\widehat{\alpha}^-} \vdash [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]N$.

Lemma 53 (Variable de-algorithmization agrees with well-formedness).

+ $T\,;\overrightarrow{\widehat{\alpha}^-} \vdash P$ *implies* $T, \overrightarrow{\alpha^-} \vdash [\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}]P$;
− $T\,;\overrightarrow{\widehat{\alpha}^-} \vdash N$ *implies* $T, \overrightarrow{\alpha^-} \vdash [\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}]N$.

Corollary 22 (Well-formedness Algorithmic Context Weakening). *Suppose that* $T_1 \subseteq T_2$, *and* $\Upsilon_1 \subseteq \Upsilon_2$. *Then*

+ *if* $T_1\,;\Upsilon_1 \vdash P$ *implies* $T_2\,;\Upsilon_2 \vdash P$,
− *if* $T_1\,;\Upsilon_1 \vdash N$ *implies* $T_2\,;\Upsilon_2 \vdash N$.

7.2 Algorithmic Substitution

Lemma 54 (Determinacy of typing algorithm). *Suppose that* $T \vdash \Gamma$ *and* $T \vdash^{\supseteq} \Sigma$. *Then*

+ *If* $T;\Gamma \vDash v\colon P$ *and* $T;\Gamma \vDash v\colon P'$ *then* $P = P'$.
− *If* $T;\Gamma \vDash c\colon N$ *and* $T;\Gamma \vDash c\colon N'$ *then* $N = N'$.
• *If* $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M \dashv \Sigma'\,;C$ *and* $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M' \dashv \Sigma'\,;C'$ *then* $M = M'$, $\Sigma = \Sigma'$, *and* $C = C'$.

Lemma 55 (Algorithmic Substitution Strengthening). *Restricting the substitution to the algorithmic variables of the substitution subject does not affect the result. Suppose that* $\widehat{\sigma}$ *is an algorithmic substitution,* P *and* N *are algorithmic types. Then*

+ $[\widehat{\sigma}]P = [\widehat{\sigma}|_{\mathbf{fav}\,P}]P$,

– $[\widehat{\sigma}]N = [\widehat{\sigma}|_{\mathbf{fav}\,N}]N$

Lemma 56 (Substitutions equal on the algorithmic variables). *Suppose that $\widehat{\sigma}_1$ and $\widehat{\sigma}_2$ are normalized substitutions of signature $\Sigma \vdash \widehat{\sigma}_i : \Upsilon$. Then*

+ *for a normalized type $T\,;\Upsilon \vdash P$, if $[\widehat{\sigma}_1]P = [\widehat{\sigma}_2]P$ then $\widehat{\sigma}_1|_{(\mathbf{fav}\,P)} = \widehat{\sigma}_2|_{(\mathbf{fav}\,P)}$;*
– *for a normalized type $T\,;\Upsilon \vdash N$, if $[\widehat{\sigma}_1]N = [\widehat{\sigma}_2]N$ then $\widehat{\sigma}_1|_{(\mathbf{fav}\,N)} = \widehat{\sigma}_2|_{(\mathbf{fav}\,N)}$.*

Corollary 23 (Substitutions equivalent on the algorithmic variables). *Suppose that $\widehat{\sigma}_1$ and $\widehat{\sigma}_2$ are substitutions of signature $\Sigma \vdash \widehat{\sigma}_i : \Upsilon$ where $T \vdash^{\supseteq} \Sigma$. Then*

+ *for a type $T\,;\Upsilon \vdash P$, if $T \vdash [\widehat{\sigma}_1]P \simeq^{\leqslant} [\widehat{\sigma}_2]P$ then $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{fav}\,P$;*
– *for a type $T\,;\Upsilon \vdash N$, if $T \vdash [\widehat{\sigma}_1]N \simeq^{\leqslant} [\widehat{\sigma}_2]N$ then $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{fav}\,N$.*

7.3 Algorithmic Normalization

Lemma 57 (Determinacy of typing algorithm). *Suppose that $T \vdash \Gamma$ and $T \vdash^{\supseteq} \Sigma$. Then*

+ *If $T;\Gamma \vDash v\colon P$ and $T;\Gamma \vDash v\colon P'$ then $P = P'$.*
– *If $T;\Gamma \vDash c\colon N$ and $T;\Gamma \vDash c\colon N'$ then $N = N'$.*
• *If $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M \dashv \Sigma'\,;C$ and $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M' \dashv \Sigma'\,;C'$ then $M = M'$, $\Sigma = \Sigma'$, and $C = C'$.*

Lemma 58 (Algorithmic variables are not changed by the normalization).

– $\mathbf{fav}\,N \equiv \mathbf{favnf}\,(N)$
+ $\mathbf{fav}\,P \equiv \mathbf{favnf}\,(P)$

Lemma 59 (Soundness of normalization of algorithmic types).

– $N \simeq^D \mathbf{nf}\,(N)$
+ $P \simeq^D \mathbf{nf}\,(P)$

7.4 Algorithmic Equivalence

Lemma 60 (Algorithmic type well-formedness is invariant under equivalence). *Mutual subtyping implies declarative equivalence.*

+ *if $P \simeq^D Q$ then $T\,;\Upsilon \vdash P \iff T\,;\Upsilon \vdash Q$,*
– *if $N \simeq^D M$ then $T\,;\Upsilon \vdash N \iff T\,;\Upsilon \vdash M$*

Corollary 24 (Normalization preserves well-formedness of algorithmic types).

+ $T\,;\Upsilon \vdash P \iff T\,;\Upsilon \vdash \mathbf{nf}\,(P)$,
– $T\,;\Upsilon \vdash N \iff T\,;\Upsilon \vdash \mathbf{nf}\,(N)$

Corollary 25 (Normalization preserves the signature of the algorithmic substitution). $\Sigma \vdash \widehat{\sigma} : \Upsilon \iff \Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}) : \Upsilon, T \vdash \widehat{\sigma} : \Upsilon \iff T \vdash \mathbf{nf}\,(\widehat{\sigma}) : \Upsilon$.

Corollary 26 (Algorithmic substitution equivalence becomes equality after normalization). *Suppose that $\Sigma \vdash \widehat{\sigma}_1 : \Upsilon'$ and $\Sigma \vdash \widehat{\sigma}_2 : \Upsilon'$ are algorithmic substitutions and $\Upsilon \subseteq \Upsilon'$. Then $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \Upsilon \iff$* $\mathbf{nf}\,(\widehat{\sigma}_1)|_{\Upsilon} = \mathbf{nf}\,(\widehat{\sigma}_2)|_{\Upsilon}$.

7.5 Unification Constraint Merge

Observation 3 (Unification Constraint Merge Determinism). *Suppose that $\Sigma \vdash UC_1$ and $\Sigma \vdash UC_2$ If $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$ and $\Sigma \vdash UC_1 \,\&\, UC_2 = UC'$ are defined then $UC = UC'$.*

Lemma 61 (Soundness of Unification Constraint Merge). *Suppose that $\Sigma \vdash UC_1$ and $\Sigma \vdash UC_2$ are normalized unification constraints. If $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$ is defined then $UC = UC_1 \cup UC_2$.*

Corollary 27. *Suppose that $\Sigma \vdash UC_1$ and $\Sigma \vdash UC_2$ are normalized unification constraints. If $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$ is defined then*

(1) $\Sigma \vdash UC$ is normalized unification constraint,
(2) for any substitution $\Sigma \vdash \widehat{\sigma} : \mathbf{dom}\,(UC)$, $\Sigma \vdash \widehat{\sigma} : UC$ implies $\Sigma \vdash \widehat{\sigma} : UC_1$ and $\Sigma \vdash \widehat{\sigma} : UC_2$.

Lemma 62 (Completeness of Unification Constraint Entry Merge). *For a fixed context T, suppose that $T \vdash ue_1$ and $T \vdash ue_2$ are matching constraint entries.*

+ *for a type P such that $T \vdash P : ue_1$ and $T \vdash P : ue_2$, $T \vdash ue_1 \,\&\, ue_2 = ue$ is defined and $T \vdash P : ue$.*
− *for a type N such that $T \vdash N : ue_1$ and $T \vdash N : ue_2$, $T \vdash ue_1 \,\&\, ue_2 = ue$ is defined and $T \vdash N : ue$.*

Lemma 63 (Completeness of Unification Constraint Merge). *Suppose that $\Sigma \vdash UC_1$ and $\Sigma \vdash UC_2$. Then for any $\Upsilon \supseteq \mathbf{dom}\,(UC_1) \cup \mathbf{dom}\,(UC_2)$ and substitution $\Sigma \vdash \widehat{\sigma} : \Upsilon$ such that $\Sigma \vdash \widehat{\sigma} : UC_1$ and $\Sigma \vdash \widehat{\sigma} : UC_2$,*

(1) $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$ is defined and
(2) $\Sigma \vdash \widehat{\sigma} : UC$.

7.6 Unification

Observation 4 (Unification Determinism).

+ *If $T; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$ and $T; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC'$ then $UC = UC'$.*
− *If $T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC$ and $T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC'$ then $UC = UC'$.*

Lemma 64 (Soundness of Unification).

+ *For normalized P and Q such that $T\,;\mathbf{dom}\,(\Sigma) \vdash P$ and $T \vdash Q$, if $T; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$ then $\Sigma \vdash UC : \mathbf{fav} P$ and for any normalized $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : UC$, $[\widehat{\sigma}]P = Q$.*
− *For normalized N and M such that $T\,;\mathbf{dom}\,(\Sigma) \vdash N$ and $T \vdash M$, if $T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC$ then $\Sigma \vdash UC : \mathbf{fav} N$ and for any normalized $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : UC$, $[\widehat{\sigma}]N = M$.*

Lemma 65 (Completeness of Unification).

+ *For normalized P and Q such that $T\,;\mathbf{dom}\,(\Sigma) \vdash P$ and $T \vdash Q$, suppose that there exists $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P)$ such that $[\widehat{\sigma}]P = Q$, then $T; \Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$ for some UC.*
− *For normalized N and M such that $T\,;\mathbf{dom}\,(\Sigma) \vdash N$ and $T \vdash M$, suppose that there exists $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N)$ such that $[\widehat{\sigma}]N = M$, then $T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC$ for some UC.*

7.7 Anti-unification

Observation 5 (Determinism of Anti-unification Algorithm).

+ *If* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ *and* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$*, then* $\Upsilon = \Upsilon'$, $Q = Q'$, $\widehat{\tau}_1 = \widehat{\tau}'_1$, *and* $\widehat{\tau}_2 = \widehat{\tau}'_2$.
− *If* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ *and* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon', M', \widehat{\tau}'_1, \widehat{\tau}'_2)$*, then* $\Upsilon = \Upsilon'$, $M = M'$, $\widehat{\tau}_1 = \widehat{\tau}'_1$, *and* $\widehat{\tau}_2 = \widehat{\tau}'_2$.

Observation 6 (Uniqueness of Anti-unification Variable Names). *Names of the anti-unification variables are uniquely defined by the types they are mapped to by the resulting substitutions.*

+ *Assuming* P_1 *and* P_2 *are normalized, if* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ *then for any* $\widehat{\beta}^{-} \in \Upsilon$, $\widehat{\beta}^{-} = \widehat{\alpha}^{-}_{\{[\widehat{\tau}_1]\widehat{\beta}^{-}, [\widehat{\tau}_2]\widehat{\beta}^{-}\}}$
− *Assuming* N_1 *and* N_2 *are normalized, if* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ *then for any* $\widehat{\beta}^{-} \in \Upsilon$, $\widehat{\beta}^{-} = \widehat{\alpha}^{-}_{\{[\widehat{\tau}_1]\widehat{\beta}^{-}, [\widehat{\tau}_2]\widehat{\beta}^{-}\}}$

Lemma 66 (Soundness of Anti-Unification).

+ *Assuming* P_1 *and* P_2 *are normalized, if* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ *then*
  (1) $T ; \Upsilon \vdash Q$,
  (2) $T ; \cdot \vdash \widehat{\tau}_i : \Upsilon$ *for* $i \in \{1, 2\}$ *are anti-unification substitutions, and*
  (3) $[\widehat{\tau}_i]Q = P_i$ *for* $i \in \{1, 2\}$.
− *Assuming* N_1 *and* N_2 *are normalized, if* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ *then*
  (1) $T ; \Upsilon \vdash M$,
  (2) $T ; \cdot \vdash \widehat{\tau}_i : \Upsilon$ *for* $i \in \{1, 2\}$ *are anti-unification substitutions, and*
  (3) $[\widehat{\tau}_i]M = N_i$ *for* $i \in \{1, 2\}$.

Lemma 67 (Completeness of Anti-Unification).

+ *Assume that* P_1 *and* P_2 *are normalized, and there exists* $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$ *such that*
  (1) $T ; \Upsilon' \vdash Q'$,
  (2) $T ; \cdot \vdash \widehat{\tau}'_i : \Upsilon'$ *for* $i \in \{1, 2\}$ *are anti-unification substitutions, and*
  (3) $[\widehat{\tau}'_i]Q' = P_i$ *for* $i \in \{1, 2\}$.
  Then the anti-unification algorithm terminates, that is there exists $(\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ *such that* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$
− *Assume that* N_1 *and* N_2 *are normalized, and there exists* $(\Upsilon', M', \widehat{\tau}'_1, \widehat{\tau}'_2)$ *such that*
  (1) $T ; \Upsilon' \vdash M'$,
  (2) $T ; \cdot \vdash \widehat{\tau}'_i : \Upsilon'$ *for* $i \in \{1, 2\}$*, are anti-unification substitutions, and*
  (3) $[\widehat{\tau}'_i]M' = N_i$ *for* $i \in \{1, 2\}$.
  Then the anti-unification algorithm succeeds, that is there exists $(\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ *such that* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$.

Lemma 68 (Initiality of Anti-Unification).

+ *Assume that* P_1 *and* P_2 *are normalized, and* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$*, then* $(\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ *is more specific than any other sound anti-unifier* $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$*, i.e. if*
  (1) $T ; \Upsilon' \vdash Q'$,
  (2) $T ; \cdot \vdash \widehat{\tau}'_i : \Upsilon'$ *for* $i \in \{1, 2\}$*, and*

(3) $[\widehat{\tau}'_i]\, Q' = P_i$ *for* $i \in \{1, 2\}$

then there exists $\widehat{\rho}$ *such that* $T\,;\Upsilon \vdash \widehat{\rho} : (\Upsilon'|_{\mathbf{fv}\, Q'})$ *and* $[\widehat{\rho}]\, Q' = Q$. *Moreover,* $[\widehat{\rho}]\widehat{\beta}^{-}$ *can be uniquely determined by* $[\widehat{\tau}'_1]\widehat{\beta}^{-}$, $[\widehat{\tau}'_2]\widehat{\beta}^{-}$, *and* T.

- *Assume that* N_1 *and* N_2 *are normalized, and* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$*, then* $(\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ *is more specific than any other sound anti-unifier* $(\Upsilon', M', \widehat{\tau}'_1, \widehat{\tau}'_2)$*, i.e. if*
  (1) $T\,;\Upsilon' \vdash M'$,
  (2) $T\,; \cdot \vdash \widehat{\tau}'_i : \Upsilon'$ *for* $i \in \{1, 2\}$*, and*
  (3) $[\widehat{\tau}'_i]\, M' = N_i$ *for* $i \in \{1, 2\}$

  then there exists $\widehat{\rho}$ *such that* $T\,;\Upsilon \vdash \widehat{\rho} : (\Upsilon'|_{\mathbf{fv}\, M'})$ *and* $[\widehat{\rho}]\, M' = M$. *Moreover,* $[\widehat{\rho}]\widehat{\beta}^{-}$ *can be uniquely determined by* $[\widehat{\tau}'_1]\widehat{\beta}^{-}$, $[\widehat{\tau}'_2]\widehat{\beta}^{-}$, *and* T.

7.8 Upper Bounds

Observation 7 (Determinism of Least Upper Bound algorithm). *For types* $T \vdash P_1$*, and* $T \vdash P_2$*, if* $T \vDash P_1 \vee P_2 = Q$ *and* $T \vDash P_1 \vee P_2 = Q'$ *then* $Q = Q'$.

Lemma 69 (Characterization of the Supertypes). *Let us define the set of upper bounds of a positive type* $\mathsf{UB}(P)$ *in the following way:*

$T \vdash P$	$\mathsf{UB}(T \vdash P)$
$T \vdash \beta^{+}$	$\{\exists \overrightarrow{\alpha^{-}}.\, \beta^{+} \mid \textit{for } \overrightarrow{\alpha^{-}}\}$
$T \vdash \exists \overrightarrow{\beta^{-}}.\, Q$	$\mathsf{UB}(T, \overrightarrow{\beta^{-}} \vdash Q)$ *not using* $\overrightarrow{\beta^{-}}$
$T \vdash {\downarrow} M$	$\left\{ \exists \overrightarrow{\alpha^{-}}.\, {\downarrow} M' \;\middle\vert\; \begin{array}{l} \textit{for } \overrightarrow{\alpha^{-}},\, M', \textit{ and } \overrightarrow{N} \textit{ s.t.} \\ T \vdash N_i,\, T, \overrightarrow{\alpha^{-}} \vdash M', \textit{ and } [\overrightarrow{N}/\overrightarrow{\alpha^{-}}]{\downarrow} M' \simeq^{D} {\downarrow} M \end{array} \right\}$

Then $\mathsf{UB}(T \vdash P) \equiv \{Q \mid T \vdash Q \geqslant P\}$.

Lemma 70 (Characterization of the Normalized Supertypes). *For a normalized positive type* $P = \mathbf{nf}\,(P)$*, let us define the set of normalized upper bounds in the following way:*

$T \vdash P$	$\mathsf{NFUB}(T \vdash P)$
$T \vdash \beta^{+}$	$\{\beta^{+}\}$
$T \vdash \exists \overrightarrow{\beta^{-}}.\, P$	$\mathsf{NFUB}(T, \overrightarrow{\beta^{-}} \vdash P)$ *not using* $\overrightarrow{\beta^{-}}$
$T \vdash {\downarrow} M$	$\left\{ \exists \overrightarrow{\alpha^{-}}.\, {\downarrow} M' \;\middle\vert\; \begin{array}{l} \textit{for } \overrightarrow{\alpha^{-}},\, M', \textit{ and } \overrightarrow{N} \textit{ s.t. } \mathbf{ord}\, \overrightarrow{\alpha^{-}}\, \mathbf{in}\, M' = \overrightarrow{\alpha^{-}}, \\ T \vdash N_i,\, T, \overrightarrow{\alpha^{-}} \vdash M', \textit{ and } [\overrightarrow{N}/\overrightarrow{\alpha^{-}}]{\downarrow} M' = {\downarrow} M \end{array} \right\}$

Then $\mathsf{NFUB}(T \vdash P) \equiv \{\mathbf{nf}\,(Q) \mid T \vdash Q \geqslant P\}$.

Lemma 71. *Upper bounds of a type do not depend on the context as soon as the type is well-formed in it.*

If $T_1 \vdash P$ *and* $T_2 \vdash P$ *then* $\mathsf{UB}(T_1 \vdash P) = \mathsf{UB}(T_2 \vdash P)$ *and* $\mathsf{NFUB}(T_1 \vdash P) = \mathsf{NFUB}(T_2 \vdash P)$

Lemma 72 (Soundness of the Least Upper Bound). *For types* $T \vdash P_1$*, and* $T \vdash P_2$*, if* $T \vDash P_1 \vee P_2 = Q$ *then*

(i) $T \vdash Q$

(ii) $T \vdash Q \geqslant P_1$ and $T \vdash Q \geqslant P_2$

Lemma 73 (Completeness and Initiality of the Least Upper Bound). *For types $T \vdash P_1$, $T \vdash P_2$, and $T \vdash Q$ such that $T \vdash Q \geqslant P_1$ and $T \vdash Q \geqslant P_2$, there exists Q' s.t. $T \vDash P_1 \vee P_2 = Q'$ and $T \vdash Q \geqslant Q'$.*

7.9 Upgrade

Observation 8 (Upgrade determinism). *Assuming P is well-formed in $T \subseteq \Theta$, if $\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q$ and $\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q'$ are defined then $Q = Q'$.*

Lemma 74 (Soundness of Upgrade). *Assuming P is well-formed in $T = \Theta, \overrightarrow{\alpha^{\pm}}$, if $\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q$ then*

(1) $\Theta \vdash Q$
(2) $T \vdash Q \geqslant P$

Lemma 75 (Completeness and Initiality of Upgrade). *The upgrade returns the least T-supertype of P well-formed in Θ. Assuming P is well-formed in $T = \Theta, \overrightarrow{\alpha^{\pm}}$,*
For any Q' such that

(1) $\Theta \vdash Q'$ and
(2) $T \vdash Q' \geqslant P$,

the result of the upgrade algorithm Q exists ($\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q$) and satisfies $\Theta \vdash Q' \geqslant Q$.

7.10 Constraint Satisfaction

Lemma 76 (Any constraint is satisfiable). *Suppose that $\Sigma \vdash C$ and Υ is a set such that $\mathbf{dom}\,(C) \subseteq \Upsilon \subseteq \mathbf{dom}\,(\Sigma)$. Then there exists $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : \Upsilon$ and $\Sigma \vdash \widehat{\sigma} : C$.*

Lemma 77 (Constraint Entry Satisfaction is Stable under Equivalence).

- *If $T \vdash N_1 : e$ and $T \vdash N_1 \simeq^{\leqslant} N_2$ then $T \vdash N_2 : e$.*
+ *If $T \vdash P_1 : e$ and $T \vdash P_1 \simeq^{\leqslant} P_2$ then $T \vdash P_2 : e$.*

Corollary 28 (Constraint Satisfaction is stable under Equivalence).
If $\Sigma \vdash \widehat{\sigma}_1 : C$ and $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{dom}\,(C)$ then $\Sigma \vdash \widehat{\sigma}_2 : C$;
if $\Sigma \vdash \widehat{\sigma}_1 : UC$ and $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{dom}\,(C)$ then $\Sigma \vdash \widehat{\sigma}_2 : UC$.

Corollary 29 (Normalization preserves Constraint Satisfaction).
If $\Sigma \vdash \widehat{\sigma} : C$ then $\Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}) : C$;
if $\Sigma \vdash \widehat{\sigma} : UC$ then $\Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}) : UC$.

7.11 Positive Subtyping

Observation 9 (Positive Subtyping is Deterministic). *For fixed T, Σ, P, and Q, if $T; \Sigma \vDash P \geqslant Q \dashv C$ and $T; \Sigma \vDash P \geqslant Q \dashv C'$ then $C = C'$.*

Lemma 78 (Soundness of the Positive Subtyping). *If $T \vdash^{\supseteq} \Sigma$, $T \vdash Q$, $T\,;\mathbf{dom}\,(\Sigma) \vdash P$, and $T; \Sigma \vDash P \geqslant Q \dashv C$, then $\Sigma \vdash C : \mathbf{fav}\, P$ and for any normalized $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : C$, $T \vdash [\widehat{\sigma}]P \geqslant Q$.*

Lemma 79 (Completeness of the Positive Subtyping). *Suppose that $T \vdash^{\supseteq} \Sigma$, $T \vdash Q$ and $T\,;\mathbf{dom}\,(\Sigma) \vdash P$. Then for any $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P)$ such that $T \vdash [\widehat{\sigma}]P \geqslant Q$, there exists $T; \Sigma \vDash P \geqslant Q \dashv C$ and moreover, $\Sigma \vdash \widehat{\sigma} : C$.*

7.12 Subtyping Constraint Merge

Observation 10 (Positive Subtyping is Deterministic). *For fixed T, Σ, P, and Q, if $T;\ \Sigma \vDash P \geqslant Q \dashv C$ and $T;\ \Sigma \vDash P \geqslant Q \dashv C'$ then $C = C'$.*

Observation 11 (Constraint Entry Merge is Deterministic). *For fixed T, e_1, e_2, if $T \vdash e_1 \mathbin{\&} e_2 = e$ and $T \vdash e_1 \mathbin{\&} e_2 = e'$ then $e = e'$.*

Observation 12 (Subtyping Constraint Merge is Deterministic). *Suppose that $\Sigma \vdash C_1$ and $\Sigma \vdash C_2$ If $\Sigma \vdash C_1 \mathbin{\&} C_2 = C$ and $\Sigma \vdash C_1 \mathbin{\&} C_2 = C'$ are defined then $C = C'$.*

Lemma 80 (Soundness of Constraint Entry Merge). *For a fixed context T, suppose that $T \vdash e_1$ and $T \vdash e_2$. If $T \vdash e_1 \mathbin{\&} e_2 = e$ is defined then*

(1) $T \vdash e$
(2) For any $T \vdash P$, $T \vdash P : e$ implies $T \vdash P : e_1$ and $T \vdash P : e_2$

Lemma 81 (Soundness of Constraint Merge). *Suppose that $\Sigma \vdash C_1 : \Upsilon_1$ and $\Sigma \vdash C_2 : \Upsilon_2$ and $\Sigma \vdash C_1 \mathbin{\&} C_2 = C$ is defined. Then*

(1) $\Sigma \vdash C : \Upsilon_1 \cup \Upsilon_2$,
(2) for any substitution $\Sigma \vdash \widehat{\sigma} : \Upsilon_1 \cup \Upsilon_2$, $\Sigma \vdash \widehat{\sigma} : C$ implies $\Sigma \vdash \widehat{\sigma} : C_1$ and $\Sigma \vdash \widehat{\sigma} : C_2$.

Lemma 82 (Completeness of Constraint Entry Merge). *For a fixed context T, suppose that $T \vdash e_1$ and $T \vdash e_2$ are matching constraint entries.*

- *for a type P such that $T \vdash P : e_1$ and $T \vdash P : e_2$, $T \vdash e_1 \mathbin{\&} e_2 = e$ is defined and $T \vdash P : e$.*
- *for a type N such that $T \vdash N : e_1$ and $T \vdash N : e_2$, $T \vdash e_1 \mathbin{\&} e_2 = e$ is defined and $T \vdash N : e$.*

Lemma 83 (Completeness of Constraint Merge). *Suppose that $\Sigma \vdash C_1 : \Upsilon_1$ and $\Sigma \vdash C_2 : \Upsilon_2$. If there exists a substitution $\Sigma \vdash \widehat{\sigma} : \Upsilon_1 \cup \Upsilon_2$ such that $\Sigma \vdash \widehat{\sigma} : C_1$ and $\Sigma \vdash \widehat{\sigma} : C_2$ then $\Sigma \vdash C_1 \mathbin{\&} C_2 = C$ is defined.*

7.13 Negative Subtyping

Observation 13 (Negative Algorithmic Subtyping is Deterministic). *For fixed T, Σ, M, and N, if $T;\ \Sigma \vDash N \leqslant M \dashv C$ and $T;\ \Sigma \vDash N \leqslant M \dashv C'$ then $C = C'$.*

Lemma 84 (Soundness of Negative Subtyping). *If $T \vdash^{\supseteq} \Sigma$, $T \vdash M$, $T\ ;\ \mathbf{dom}\,(\Sigma) \vdash N$ and $T;\ \Sigma \vDash N \leqslant M \dashv C$, then $\Sigma \vdash C : \mathbf{fav}(N)$ and for any normalized $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : C$, $T \vdash [\widehat{\sigma}]N \leqslant M$.*

Lemma 85 (Completeness of the Negative Subtyping). *Suppose that $T \vdash^{\supseteq} \Sigma$, $T \vdash M$, $T;\mathbf{dom}\,(\Sigma) \vdash N$, and N does not contain negative unification variables ($\widehat{\alpha}^- \notin \mathbf{fav}\, N$). Then for any $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N)$ such that $T \vdash [\widehat{\sigma}]N \leqslant M$, there exists $T;\ \Sigma \vDash N \leqslant M \dashv C$ and moreover, $\Sigma \vdash \widehat{\sigma} : C$.*

7.14 Singularity and Minimal Instantiation

Lemma 86 (Soundness of Minimal Instantiation). *Suppose that $T \vdash^{\supseteq} \Sigma$, $\Sigma \vdash C$, and $T\ ;\mathbf{dom}\,(\Sigma) \vdash P$. If P is C-minimized by $\widehat{\sigma}$ then*

- *$\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\, P$,*
- *$\Sigma \vdash \widehat{\sigma} : C$,*
- *$\widehat{\sigma}$ is normalized, and*
- *for any other $\Sigma \vdash \widehat{\sigma}' : \mathbf{fav}\, P$ respecting C (i.e., $\Sigma \vdash \widehat{\sigma}' : C$), we have $T \vdash [\widehat{\sigma}']P \geqslant [\widehat{\sigma}]P$.*

Lemma 87 (Completeness of Minimal Instantiation). *Suppose that $T \vdash^{\supseteq} \Sigma$, $\Sigma \vdash C$, $T\,;\mathbf{dom}\,(\Sigma) \vdash P$, and there exists $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\,P$ respecting C ($\Sigma \vdash \widehat{\sigma} : C$) such that for any other $\Sigma \vdash \widehat{\sigma}' : \mathbf{fav}\,P$ respecting C ($\Sigma \vdash \widehat{\sigma}' : C$), we have $T \vdash [\widehat{\sigma}']\,P \geqslant [\widehat{\sigma}]\,P$. Then P is C-minimized by $\mathbf{nf}\,(\widehat{\sigma})$.*

Observation 14 (Minimal Instantiation is Deterministic). *Suppose that $T \vdash^{\supseteq} \Sigma$, $\Sigma \vdash C$, $T\,;\mathbf{dom}\,(\Sigma) \vdash P$. Then P is C-minimized by $\widehat{\sigma}$ and P is C-minimized by $\widehat{\sigma}'$ implies $\widehat{\sigma} = \widehat{\sigma}'$.*

Lemma 88 (Soundness of Entry Singularity).

+ *Suppose e **singular with** P for P well-formed in T. Then $T \vdash P : e$, P is normalized, and for any $T \vdash P'$ such that $T \vdash P' : e$, $T \vdash P' \simeq^{\leqslant} P$;*
− *Suppose e **singular with** N for N well-formed in T. Then $T \vdash N : e$, N is normalized, and for any $T \vdash N'$ such that $T \vdash N' : e$, $T \vdash N' \simeq^{\leqslant} N$.*

Lemma 89 (Completeness of Entry Singularity).

− *Suppose that there exists N well-formed in T such that for any N' well-formed in T, $T \vdash N' : e$ implies $T \vdash N' \simeq^{\leqslant} N$. Then e **singular with nf** (N).*
+ *Suppose that there exists P well-formed in T such that for any P' well-formed in T, $T \vdash P' : e$ implies $T \vdash P' \simeq^{\leqslant} P$. Then e **singular with nf** (P) .*

Lemma 90 (Soundness of Singularity). *Suppose $\Sigma \vdash C : \Upsilon$, and C **singular with** $\widehat{\sigma}$. Then $\Sigma \vdash \widehat{\sigma} : \Upsilon$, $\Sigma \vdash \widehat{\sigma} : C$, $\widehat{\sigma}$ is normalized, and for any $\widehat{\sigma}'$ such that $\Sigma \vdash \widehat{\sigma} : C$, $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma} : \Upsilon$.*

Observation 15 (Singularity is Deterministic). *For a fixed C such that $\Sigma \vdash C : \Upsilon$, if C **singular with** $\widehat{\sigma}$ and C **singular with** $\widehat{\sigma}'$, then $\widehat{\sigma} = \widehat{\sigma}'$.*

Lemma 91 (Completeness of Singularity). *For a given $\Sigma \vdash C$, suppose that all the substitutions satisfying C are equivalent on $\Upsilon \supseteq \mathbf{dom}\,(C)$. In other words, suppose that there exists $\Sigma \vdash \widehat{\sigma}_1 : \Upsilon$ such that for any $\Sigma \vdash \widehat{\sigma} : \Upsilon$, $\Sigma \vdash \widehat{\sigma} : C$ implies $\Sigma \vdash \widehat{\sigma} \simeq^{\leqslant} \widehat{\sigma}_1 : \Upsilon$. Then*

- *C **singular with** $\widehat{\sigma}_0$ for some $\widehat{\sigma}_0$ and*
- *$\Upsilon = \mathbf{dom}\,(C)$.*

7.15 Correctness of the Typing Algorithm

Lemma 92 (Determinacy of typing algorithm). *Suppose that $T \vdash \Gamma$ and $T \vdash^{\supseteq} \Sigma$. Then*

+ *If $T; \Gamma \vDash v\colon P$ and $T; \Gamma \vDash v\colon P'$ then $P = P'$.*
− *If $T; \Gamma \vDash c\colon N$ and $T; \Gamma \vDash c\colon N'$ then $N = N'$.*
- *If $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Longrightarrow\!\!\!> M \dashv \Sigma'\,; C$ and $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Longrightarrow\!\!\!> M' \dashv \Sigma'\,; C'$ then $M = M'$, $\Sigma = \Sigma'$, and $C = C'$.*

Lemma 93 (Soundness of typing). *Suppose that $T \vdash \Gamma$. For an inference tree T_1,*

+ *If T_1 infers $T; \Gamma \vDash v\colon P$ then $T \vdash P$ and $T; \Gamma \vdash v\colon P$*
− *If T_1 infers $T; \Gamma \vDash c\colon N$ then $T \vdash N$ and $T; \Gamma \vdash c\colon N$*
- *If T_1 infers $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Longrightarrow\!\!\!> M \dashv \Sigma'\,; C$ for $T \vdash^{\supseteq} \Sigma$ and $T\,;\mathbf{dom}\,(\Sigma) \vdash N$ free from negative algorithmic variables, then*
  (1) $T \vdash^{\supseteq} \Sigma'$
  (2) $\Sigma \subseteq \Sigma'$
  (3) $T\,;\mathbf{dom}\,(\Sigma') \vdash M$
  (4) $\mathbf{dom}\,(\Sigma) \cap \mathbf{fav}(M) \subseteq \mathbf{fav}\,N$

(5) M *is normalized and free from negative algorithmic variables*
(6) $\Sigma'|_{\mathbf{fav}\,N \cup \mathbf{fav}\,M} \vdash C$
(7) *for any* $\Sigma' \vdash \widehat{\sigma} : \mathbf{fav}\,N \cup \mathbf{fav}\,M$, $\Sigma' \vdash \widehat{\sigma} : C$ *implies* $T \,;\Gamma \vdash [\widehat{\sigma}]N \bullet \vec{v} \Longrightarrow\!\!\!> [\widehat{\sigma}]M$

Lemma 94 (Completeness of Typing). *Suppose that* $T \vdash \Gamma$. *For an inference tree* T_1,

+ *If* T_1 *infers* $T;\Gamma \vdash v\colon P$ *then* $T;\Gamma \vDash v\colon \mathbf{nf}\,(P)$
− *If* T_1 *infers* $T;\Gamma \vdash c\colon N$ *then* $T;\Gamma \vDash c\colon \mathbf{nf}\,(N)$
• *If* T_1 *infers* $T\,;\Gamma \vdash [\widehat{\sigma}]N \bullet \vec{v} \Longrightarrow\!\!\!> M$ *and*
  (1) $T \vdash^{\supseteq} \Sigma$,
  (2) $T \vdash M$,
  (3) $T\,; \mathbf{dom}\,(\Sigma) \vdash N$ *(free from negative algorithmic variables, that is* $\widehat{\alpha}^- \notin \mathbf{fav}\,N$*), and*
  (4) $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N)$,
  then there exist M', Σ', *and* C *such that*
  (1) $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Longrightarrow\!\!\!> M' \dashv \Sigma'\,; C$ *and*
  (2) *for any* $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N)$ *and* $T \vdash M$ *such that* $T\,;\Gamma \vdash [\widehat{\sigma}]N \bullet \vec{v} \Longrightarrow\!\!\!> M$, *there exists* $\widehat{\sigma}'$ *such that*
    (a) $\Sigma' \vdash \widehat{\sigma}' : \mathbf{fav}\,N \cup \mathbf{fav}\,M'$ *and* $\Sigma' \vdash \widehat{\sigma}' : C$,
    (b) $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}\,N$, *and*
    (c) $T \vdash [\widehat{\sigma}']M' \simeq^{\leqslant} M$.

8 PROPERTIES OF THE DECLARATIVE TYPE SYSTEM

8.1 Type Well-Formedness

Lemma 1 (Soundness of type well-formedness).

+ *If* $T \vdash P$ *then* $\mathbf{fv}\,(P) \subseteq T$,
− *if* $T \vdash N$ *then* $\mathbf{fv}\,(N) \subseteq T$.

Proof. The proof is done by a simple structural induction on $T \vdash P$ and mutually, $T \vdash N$.

Case 1. $T \vdash \alpha^{\pm}$ means by inversion that $\alpha^{\pm} \in T$, that is, $\alpha^{\pm} = \mathbf{fv}\,(\alpha^{\pm}) \subseteq T$.
Case 2. $T \vdash Q \to M$ means by inversion that $T \vdash Q$ and $T \vdash M$. Then by the induction hypothesis, $\mathbf{fv}\,(Q) \subseteq T$ and $\mathbf{fv}\,(M) \subseteq T$, and hence, $\mathbf{fv}\,(Q \to M) = \mathbf{fv}\,(Q) \cup \mathbf{fv}\,(M) \subseteq T$.
Case 3. the cases when $P = {\downarrow}N'$ or $N = {\uparrow}P'$ are proven analogously.
Case 4. $T \vdash \forall\overrightarrow{\alpha^+}.\,M$ means by inversion that $T, \overrightarrow{\alpha^+} \vdash M$. Then by the induction hypothesis, $\mathbf{fv}\,(M) \subseteq T, \overrightarrow{\alpha^+}$, and hence, $\mathbf{fv}\,(\forall\overrightarrow{\alpha^+}.\,M) = \mathbf{fv}\,(M) \setminus \overrightarrow{\alpha^+} \subseteq T, \overrightarrow{\alpha^+} \setminus \overrightarrow{\alpha^+} = T$.
Case 5. The case $P = \exists\overrightarrow{\alpha^-}.\,Q$ is proven analogously.

□

Lemma 2 (Completeness of type well-formedness). *In the well-formedness judgment, only used variables matter:*

+ *if* $T_1 \cap \mathbf{fv}\,P = T_2 \cap \mathbf{fv}\,P$ *then* $T_1 \vdash P \iff T_2 \vdash P$,
− *if* $T_1 \cap \mathbf{fv}\,N = T_2 \cap \mathbf{fv}\,N$ *then* $T_1 \vdash N \iff T_2 \vdash N$.

Proof. By simple mutual induction on P and N. □

Corollary 1 (Context Strengthening).

+ *If* $T \vdash P$ *then* $\mathbf{fv}\,(P) \vdash P$;
− *If* $T \vdash N$ *then* $\mathbf{fv}\,(N) \vdash N$.

Proof. It follows from lemma 2 and lemma 1.

+ By lemma 1, $\mathbf{fv}\,(P) \subseteq T$, and hence, $T \cap \mathbf{fv}\,P = \mathbf{fv}\,P$, which makes lemma 2 applicable fore contexts T and $\mathbf{fv}\,(P)$.
− The negative case is proven analogously.

□

Corollary 2 (Well-formedness Context Weakening). *Suppose that $T_1 \subseteq T_2$, then*

+ *if $T_1 \vdash P$ then $T_2 \vdash P$,*
− *if $T_1 \vdash N$ then $T_2 \vdash N$.*

Proof. By lemma 1, $T_1 \vdash P$ implies $\mathbf{fv}\,(P) \subseteq T_1$, which means that $\mathbf{fv}\,(P) \subseteq T_2$, and thus, $\mathbf{fv}\,(P) = \mathbf{fv}\,(P) \cap T_1 = \mathbf{fv}\,(P) \cap T_2$. Then by lemma 2, $T_2 \vdash P$. The negative case is symmetric. □

Lemma 3 (Well-formedness agrees with substitution). *Suppose that $T_2 \vdash \sigma : T_1$. Then*

+ *$T, T_1 \vdash P$ implies $T, T_2 \vdash [\sigma]P$, and*
− *$T, T_1 \vdash N$ implies $T, T_2 \vdash [\sigma]N$.*

Proof. We prove it by induction on $T, T_1 \vdash P$ and mutually, on $T, T_1 \vdash N$. Let us consider the last rule used in the derivation.

Case 1. (Var_+^{WF}), i.e. P is α^+.
By inversion, $\alpha^+ \in T, T_1$, then
- if $\alpha^+ \in T_1$ then $T_2 \vdash [\sigma]\alpha^+$, and by weakening (corollary 2), $T, T_2 \vdash [\sigma]\alpha^+$;
- if $\alpha^+ \in T \setminus T_1$ then $[\sigma]\alpha^+ = \alpha^+$, and by ($\text{Var}_+^{\text{WF}}$), $T, T_2 \vdash \alpha^+$.

Case 2. ($\uparrow^{\text{WF}}$), i.e. P is $\downarrow N$.
Then $T, T_1 \vdash \downarrow N$ means $T, T_1 \vdash N$ by inversion, and by the induction hypothesis, $T, T_2 \vdash [\sigma]N$. Then by ($\uparrow^{\text{WF}}$), $T, T_2 \vdash \downarrow[\sigma]N$, which by definition of substitution is rewritten as $T, T_2 \vdash [\sigma]\downarrow N$.

Case 3. ($\exists^{\text{WF}}$), i.e. P is $\exists\overrightarrow{\alpha^-}.\,Q$.
Then $T, T_1 \vdash \exists\overrightarrow{\alpha^-}.\,Q$ means $T, \overrightarrow{\alpha^-}, T_1 \vdash Q$ by inversion, and by the induction hypothesis, $T, \overrightarrow{\alpha^-}, T_2 \vdash [\sigma]Q$. Then by ($\exists^{\text{WF}}$), $T, \overrightarrow{\alpha^-}, T_2 \vdash \exists\overrightarrow{\alpha^-}.\,[\sigma]Q$, which by definition of substitution is rewritten as $T, T_2 \vdash [\sigma]\exists\overrightarrow{\alpha^-}.\,Q$.

Case 4. The negative cases are proved symmetrically.

□

8.2 Substitution

Lemma 4 (Substitution strengthening). *Restricting the substitution to the free variables of the substitution subject does not affect the result. Suppose that σ is a substitution, P and N are types. Then*

+ $[\sigma]P = [\sigma|_{\mathbf{fv}\,P}]P$,
− $[\sigma]N = [\sigma|_{\mathbf{fv}\,N}]N$

Proof. First, we strengthen the statement by saying that one can restrict the substitution to an arbitrary superset of the free variables of the substitution subject:

+ $[\sigma]P = [\sigma|_{vars}]P$, for any $vars \supseteq \mathbf{fv}\,P$, and
− $[\sigma]N = [\sigma|_{vars}]N$, for any $vars \supseteq \mathbf{fv}\,N$.

Then the proof is a straightforward induction on P and mutually, on N. For the base cases:

Case 1. $N = \alpha^-$
Then $[\sigma]\alpha^- = \sigma|_{vars}(\alpha^-)$ by definition, since $\alpha^- \in \mathbf{fv}\,\alpha^- \subseteq vars$.

Case 2. $N = P \to M$
Then $[\sigma](P \to M) = [\sigma]P \to [\sigma]M$ by definition. Since $\mathbf{fv}\,P \subseteq \mathbf{fv}\,(P \to M) \subseteq vars$, the

induction hypothesis is applicable to $[\sigma]P$: $[\sigma]P = [\sigma|_{vars}]P$. Analogously, and $[\sigma]M = [\sigma|_{vars}]M$. Then $[\sigma](P \to M) = [\sigma|_{vars}]P \to [\sigma|_{vars}]M = [\sigma|_{vars}](P \to M)$.

Case 3. $N = \uparrow P$ is proved analogously to the previous case.

Case 4. $N = \forall\overrightarrow{\alpha^+}.\, M$ (where $\overrightarrow{\alpha^+}$ is not empty)

Then $[\sigma]\forall\overrightarrow{\alpha^+}.\, M = \forall\overrightarrow{\alpha^+}.\, [\sigma]M$ by definition. Let us assume $\overrightarrow{\alpha^+}$ are fresh variables, it means that $\sigma(\alpha^\pm) = \alpha^\pm$ for any $\alpha^\pm \in \overrightarrow{\alpha^+}$, and thus, $[\sigma|_{vars}] = [\sigma|_{(vars \cup \overrightarrow{\alpha^+})}]$ immediately from the definition.

Since $vars \subseteq \mathbf{fv}\,(\forall\overrightarrow{\alpha^+}.\, M) = \mathbf{fv}\, M \setminus \overrightarrow{\alpha^+}$, $vars \cup \overrightarrow{\alpha^+} \subseteq \mathbf{fv}\,(M)$. Then by the induction hypothesis, $[\sigma]M = [\sigma|_{(vars \cup \overrightarrow{\alpha^+})}]M$. Finally, $[\sigma]\forall\overrightarrow{\alpha^+}.\, M = \forall\overrightarrow{\alpha^+}.\, [\sigma|_{(vars \cup \overrightarrow{\alpha^+})}]M = \forall\overrightarrow{\alpha^+}.\, [\sigma|_{vars}]M = [\sigma|_{vars}]\forall\overrightarrow{\alpha^+}.\, M$.

Case 5. The positive cases are proves symmetrically.

□

Lemma 5 (Signature of a restricted substitution). *If $T_2 \vdash \sigma : T_1$ then $T_2 \vdash \sigma|_{vars} : T_1 \cap vars$.*

Proof. Let us take an arbitrary $\alpha^\pm \in T_1 \cap vars$. Since $\alpha^\pm \in T_1$, $T_2 \vdash [\sigma]\alpha^\pm$ by the signature of σ.

Let us take an arbitrary $\alpha^\pm \notin T_1 \cap vars$. If $\alpha^\pm \notin vars$ then $[\sigma|_{vars}]\alpha^\pm = \alpha^\pm$ by definition of restriction. If $\alpha^\pm \in vars \setminus T_1$ then $[\sigma|_{vars}]\alpha^\pm = [\sigma]\alpha^\pm$ by definition and $[\sigma]\alpha^\pm = \alpha^\pm$ by the signature of σ. □

Lemma 6. *Suppose that σ is a substitution with signature $T_2 \vdash \sigma : T_1$. Then if vars is disjoint from T_1, then $\sigma|_{vars} =$ **id**.*

Proof. Let us take an arbitrary $\alpha^\pm$. If $\alpha^\pm \notin vars$ then $[\sigma|_{vars}]\alpha^\pm = \alpha^\pm$ by definition.

If $\alpha^\pm \in vars$ then $\alpha^\pm \notin T_1$ by assumption. Then $[\sigma|_{vars}]\alpha^\pm = [\sigma]\alpha^\pm$ by definition of restricted substitution, and since $T_2 \vdash \sigma : T_1$, we have $[\sigma]\alpha^\pm = \alpha^\pm$. □

Corollary 3 (Application of a disjoint substitution). *Suppose that σ is a substitution with signature $T_2 \vdash \sigma : T_1$. Then*

+ *if $T_1 \cap \mathbf{fv}\,(Q) = \emptyset$ then $[\sigma]Q = Q$;*
– *if $T_1 \cap \mathbf{fv}\,(N) = \emptyset$ then $[\sigma]N = N$.*

Lemma 7 (Substitution range weakening). *Suppose that $T_2 \subseteq T_2'$ are contexts and σ is a substitution. Then $T_2 \vdash \sigma : T_1$ implies $T_2' \vdash \sigma : T_1$.*

Proof. For any $\alpha^\pm \in T_1$, $T_2 \vdash \sigma : T_1$ gives us $T_2 \vdash [\sigma]\alpha^\pm$, which can be weakened to $T_2' \vdash [\sigma]\alpha^\pm$ by corollary 2. This way, $T_2' \vdash \sigma : T_1$. □

Lemma 8 (Subsitutions Equivalent on Free Variables). *Suppose that $T' \subseteq T$, σ_1 and σ_2 are substitutions of signature $T \vdash \sigma_i : T'$. Then*

+ *for a type $T \vdash P$, if $T \vdash [\sigma_1]P \simeq^\leqslant [\sigma_2]P$ then $T \vdash \sigma_1 \simeq^\leqslant \sigma_2 : \mathbf{fv}\, P \cap T'$;*
– *for a type $T \vdash N$, if $T \vdash [\sigma_1]N \simeq^\leqslant [\sigma_2]N$ then $T \vdash \sigma_1 \simeq^\leqslant \sigma_2 : \mathbf{fv}\, N \cap T'$.*

Proof. Let us make an additional assumption that σ_1, σ_2, and the mentioned types are normalized. If they are not, we normalize them first.

Notice that the normalization preserves the set of free variables (lemma 38), well-formedness (corollary 14), and equivalence (lemma 46), and distributes over substitution (lemma 41). This way, the assumed and desired properties are equivalent to their normalized versions.

We prove it by induction on the structure of P and mutually, N. Let us consider the shape of this type.

Case 1. $P = \alpha^+ \in T'$. Then $T \vdash \sigma_1 \simeq^\leqslant \sigma_2 : \mathbf{fv}\, P \cap T'$ means $T \vdash \sigma_1 \simeq^\leqslant \sigma_2 : \alpha^+$, i.e. $T \vdash [\sigma_1]\alpha^+ \simeq^\leqslant [\sigma_2]\alpha^+$, which holds by assumption.

Case 2. $P = \alpha^+ \in T \setminus T'$. Then $\mathbf{fv}\, P \cap T' = \emptyset$, so $T \vdash \sigma_1 \simeq^\leqslant \sigma_2 : \mathbf{fv}\, P \cap T'$ holds vacuously.

Case 3. $P = \downarrow N$. Then the induction hypothesis is applicable to type N:

(1) N is normalized,
(2) $T \vdash N$ by inversion of $T \vdash \downarrow N$,
(3) $T \vdash [\sigma_1]N \simeq^\leqslant [\sigma_2]N$ holds by inversion of $T \vdash [\sigma_1]\downarrow N \simeq^\leqslant [\sigma_2]\downarrow N$, i.e. $T \vdash \downarrow[\sigma_1]N \simeq^\leqslant \downarrow[\sigma_2]N$.

This way, we obtain $T \vdash \sigma_1 \simeq^\leqslant \sigma_2 : \mathbf{fv}\, N \cap T'$, which implies the required equivalence since $\mathbf{fv}\, P \cap T' = \mathbf{fv}\downarrow N \cap T' = \mathbf{fv}\, N \cap T'$.

Case 4. $P = \exists\overrightarrow{\alpha^-}.\, Q$ Then the induction hypothesis is applicable to type Q well-formed in context $T, \overrightarrow{\alpha^-}$:

(1) $T' \subseteq T, \overrightarrow{\alpha^-}$ since $T' \subseteq T$,
(2) $T, \overrightarrow{\alpha^-} \vdash \sigma_i : T'$ by weakening,
(3) Q is normalized,
(4) $T, \overrightarrow{\alpha^-} \vdash Q$ by inversion of $T \vdash \exists\overrightarrow{\alpha^-}.\, Q$,
(5) Notice that $[\sigma_i]\exists\overrightarrow{\alpha^-}.\, Q$ is normalized, and thus, $[\sigma_1]\exists\overrightarrow{\alpha^-}.\, Q \simeq^D [\sigma_2]\exists\overrightarrow{\alpha^-}.\, Q$ implies $[\sigma_1]\exists\overrightarrow{\alpha^-}.\, Q = [\sigma_2]\exists\overrightarrow{\alpha^-}.\, Q$ (by lemma 46).). This equality means $[\sigma_1]Q = [\sigma_2]Q$, which implies $T \vdash [\sigma_1]Q \simeq^\leqslant [\sigma_2]Q$.

Case 5. $N = P \to M$

□

Lemma 9 (Substitution composition well-formedness). *If $T_1' \vdash \sigma_1 : T_1$ and $T_2' \vdash \sigma_2 : T_2$, then $T_1', T_2' \vdash \sigma_2 \circ \sigma_1 : T_1, T_2$.*

Proof. By definition of composition. □

Lemma 10 (Substitution monadic composition well-formedness). *If $T_1' \vdash \sigma_1 : T_1$ and $T_2' \vdash \sigma_2 : T_2$, then $T_2' \vdash \sigma_2 \lll \sigma_1 : T_1$.*

Proof. By definition of monadic composition. □

Lemma 11 (Substitution composition). *If $T_1' \vdash \sigma_1 : T_1$, $T_2' \vdash \sigma_2 : T_2$, $T_1 \cap T_2' = \emptyset$ and $T_1 \cap T_2 = \emptyset$ then $\sigma_2 \circ \sigma_1 = (\sigma_2 \lll \sigma_1) \circ \sigma_2$.*

Proof.

(1) Suppose that $\alpha^\pm \in T_1$ then $\alpha^\pm \notin T_2$, and thus, $[(\sigma_2 \lll \sigma_1) \circ \sigma_2]\alpha^\pm = [(\sigma_2 \lll \sigma_1)]\alpha^\pm = [\sigma_2][\sigma_1]\alpha^\pm = [(\sigma_2 \circ \sigma_1)]\alpha^\pm$.
(2) Suppose that $\alpha^\pm \notin T_1$ then $[(\sigma_2 \circ \sigma_1)]\alpha^\pm = [\sigma_2]\alpha^\pm$. Then
    (a) if $\alpha^\pm \notin T_2$ then $[\sigma_2]\alpha^\pm = \alpha^\pm$ and $[(\sigma_2 \lll \sigma_1) \circ \sigma_2]\alpha^\pm = [(\sigma_2 \lll \sigma_1)][\sigma_2]\alpha^\pm = [\sigma_2 \lll \sigma_1]\alpha^\pm = \alpha^\pm$
    (b) if $\alpha^\pm \in T_2$ then $T_2' \vdash [\sigma_2]\alpha^\pm$, and hence, $[(\sigma_2 \lll \sigma_1) \circ \sigma_2]\alpha^\pm = [(\sigma_2 \lll \sigma_1)][\sigma_2]\alpha^\pm = [\sigma_2]\alpha^\pm$ by definition of monadic composition, since none of the free variables of $[\sigma_2]\alpha^\pm$ is in T_1.

□

Corollary 4 (Substitution composition commutativity). *If $T_1' \vdash \sigma_1 : T_1$, $T_2' \vdash \sigma_2 : T_2$, and $T_1 \cap T_2 = \emptyset$, $T_1 \cap T_2' = \emptyset$, and $T_1' \cap T_2 = \emptyset$ then $\sigma_2 \circ \sigma_1 = \sigma_1 \circ \sigma_2$.*

Proof. by lemma 11, $\sigma_2 \circ \sigma_1 = (\sigma_2 \lll \sigma_1) \circ \sigma_2$. Since the codomain of σ_1 is T_1', and it is disjoint with the domain of σ_2, $\sigma_2 \lll \sigma_1 = \sigma_1$. □

Lemma 12 (Substitution domain weakening). *If $T_2 \vdash \sigma : T_1$ then $T_2, T' \vdash \sigma : T_1, T'$*

Proof. If the variable $\alpha^\pm$ is in T_1 then $T_2 \vdash [\sigma]\alpha^\pm$ by assumption, and then $T_2, T' \vdash [\sigma]\alpha^\pm$ by weakening. If the variable $\alpha^\pm$ is in $T' \setminus T_1$ then $[\sigma]\alpha^\pm = \alpha^\pm \in T' \subseteq T_2, T'$, and thus, $T_2, T' \vdash \alpha^\pm$. □

Lemma 13 (Free variables after substitution). *Suppose that $T_2 \vdash \sigma : T_1$, then*

+ *for a type P, the free variables of $[\sigma]P$ are bounded in the following way:* $\mathbf{fv}(P) \setminus T_1 \subseteq \mathbf{fv}([\sigma]P) \subseteq (\mathbf{fv}(P) \setminus T_1) \cup T_2$*;*
− *for a type N, the free variables of $[\sigma]P$ are bounded in the following way:* $\mathbf{fv}(N) \setminus T_1 \subseteq \mathbf{fv}([\sigma]N) \subseteq (\mathbf{fv}(N) \setminus T_1) \cup T_2$*.*

Proof. We prove it by structural induction on P and mutually, on N.

Case 1. $P = \alpha^+$
If $\alpha^+ \in T_1$ then $T_2 \vdash [\sigma]\alpha^+$, and by lemma 1, $\mathbf{fv}([\sigma]\alpha^+) \subseteq T_2$. $\mathbf{fv}(\alpha^+) \setminus T_1 = \emptyset$, so $\mathbf{fv}([\sigma]P) \setminus T_1 \subseteq \mathbf{fv}([\sigma]\alpha^+)$ vacuously.
If $\alpha^+ \notin T_1$ then $[\sigma]\alpha^+ = \alpha^+$, and $\mathbf{fv}([\sigma]\alpha^+) = \alpha^+ = \alpha^+ \setminus T_1$.
Case 2. $P = \exists\overrightarrow{\alpha^-}. Q$
Then we need to show that $\mathbf{fv}([\sigma]P) = \mathbf{fv}([\sigma]Q) \setminus \overrightarrow{\alpha^-}$ is a subset of $(\mathbf{fv}(P) \setminus T_1) \cup T_2$ and a superset of $\mathbf{fv}(P) \setminus T_1$. Notice that $\mathbf{fv}(P) = \mathbf{fv}(Q) \setminus \overrightarrow{\alpha^-}$ by definition. This way, we need to show that $\mathbf{fv}(Q) \setminus \overrightarrow{\alpha^-} \setminus T_1 \subseteq \mathbf{fv}([\sigma]Q) \setminus \overrightarrow{\alpha^-} \subseteq (\mathbf{fv}(Q) \setminus \overrightarrow{\alpha^-} \setminus T_1) \cup T_2$,
By the induction hypothesis, $\mathbf{fv}([\sigma]Q) \subseteq (\mathbf{fv}(Q) \setminus T_1) \cup T_2$. So for the second inclusion, it suffices to show that $((\mathbf{fv}(Q) \setminus T_1) \cup T_2) \setminus \overrightarrow{\alpha^-} \subseteq (\mathbf{fv}(Q) \setminus \overrightarrow{\alpha^-} \setminus T_1) \cup T_2$, which holds by set theoretical reasoning.
Also by the induction hypothesis, $\mathbf{fv}(Q) \setminus T_1 \subseteq \mathbf{fv}([\sigma]Q)$, and thus, by subtracting $\overrightarrow{\alpha^-}$ from both sides, $\mathbf{fv}(Q) \setminus \overrightarrow{\alpha^-} \setminus T_1 \subseteq \mathbf{fv}([\sigma]Q) \setminus \overrightarrow{\alpha^-}$.
Case 3. The case $N = \forall\overrightarrow{\alpha^+}. M$ is proved analogously.
Case 4. $N = P \to M$
Then $\mathbf{fv}([\sigma]N) = \mathbf{fv}([\sigma]P) \cup \mathbf{fv}([\sigma]M)$. By the induction hypothesis,
(1) $\mathbf{fv}(P) \setminus T_1 \subseteq \mathbf{fv}([\sigma]P) \subseteq (\mathbf{fv}(P) \setminus T_1) \cup T_2$ and
(2) $\mathbf{fv}(M) \setminus T_1 \subseteq \mathbf{fv}([\sigma]M) \subseteq (\mathbf{fv}(M) \setminus T_1) \cup T_2$.
We unite these inclusions vertically and obtain $\mathbf{fv}(P) \setminus T_1 \cup \mathbf{fv}(M) \setminus T_1 \subseteq \mathbf{fv}([\sigma]N) \subseteq ((\mathbf{fv}(P) \setminus T_1) \cup T_2) \cup ((\mathbf{fv}(M) \setminus T_1) \cup T_2)$, which is equivalent to $(\mathbf{fv}(P) \cup \mathbf{fv}(M)) \setminus T_1 \subseteq \mathbf{fv}([\sigma]N) \subseteq (\mathbf{fv}(P) \cup \mathbf{fv}(M)) \setminus T_1 \cup T_2$. Since $\mathbf{fv}(P) \cup \mathbf{fv}(M) = \mathbf{fv}(N)$, $\mathbf{fv}(N) \setminus T_1 \subseteq \mathbf{fv}([\sigma]N) \subseteq (\mathbf{fv}(N) \setminus T_1) \cup T_2$.
Case 5. The cases when $P = \downarrow M$ and $N = \uparrow Q$ are proved analogously

□

Lemma 14 (Free variables of a variable image). *Suppose that σ is an arbitrary substitution, Then*

+ *if $\alpha^\pm \in \mathbf{fv}(P)$ then $\mathbf{fv}([\sigma]\alpha^\pm) \subseteq \mathbf{fv}([\sigma]P)$,*
− *if $\alpha^\pm \in \mathbf{fv}(N)$ then $\mathbf{fv}([\sigma]\alpha^\pm) \subseteq \mathbf{fv}([\sigma]N)$.*

Proof. By mutual induction on P and N. The base cases (when P or N is a variable) are trivial, since then $\alpha^\pm \in \mathbf{fv}(P)$ means $\alpha^\pm = P$ (and symmetrically for N). The congruent cases (when the type is formed by $\downarrow$, $\uparrow$, or $\to$) hold since $\alpha^\pm$ occurs in type means that it occurs in one of its parts, to which we apply the induction hypothesis.

Let us suppose that the type is $\exists\overrightarrow{\alpha^-}. Q$. Then $\alpha^\pm \in \mathbf{fv}(\exists\overrightarrow{\alpha^-}. Q)$ means $\alpha^\pm \in \mathbf{fv}(Q)$ and $\alpha^\pm \notin \overrightarrow{\alpha^-}$. Then by the induction hypothesis, $\mathbf{fv}([\sigma]\alpha^\pm) \subseteq \mathbf{fv}([\sigma]Q)$, and it is left to notice that $\mathbf{fv}([\sigma]\alpha^\pm) \cap \overrightarrow{\alpha^-} = \emptyset$, which we can ensure by alpha-equivalence. □

8.3 Declarative Subtyping

Lemma 15 (Free Variable Propagation). *In the judgments of negative subtyping or positive supertyping, free variables propagate left to right. For a context T,*

– *if* $T \vdash N \leqslant M$ *then* $\mathbf{fv}(N) \subseteq \mathbf{fv}(M)$
+ *if* $T \vdash P \geqslant Q$ *then* $\mathbf{fv}(P) \subseteq \mathbf{fv}(Q)$

Proof. Mutual induction on $T \vdash N \leqslant M$ and $T \vdash P \geqslant Q$.

Case 1. $T \vdash \alpha^- \leqslant \alpha^-$
It is self-evident that $\alpha^- \subseteq \alpha^-$.

Case 2. $T \vdash \uparrow P \leqslant \uparrow Q$ From the inversion (and unfolding $T \vdash P \simeq^{\leqslant} Q$), we have $T \vdash P \geqslant Q$. Then by the induction hypothesis, $\mathbf{fv}(P) \subseteq \mathbf{fv}(Q)$. The desired inclusion holds, since $\mathbf{fv}(\uparrow P) = \mathbf{fv}(P)$ and $\mathbf{fv}(\uparrow Q) = \mathbf{fv}(Q)$.

Case 3. $T \vdash P \to N \leqslant Q \to M$ The induction hypothesis applied to the premises gives: $\mathbf{fv}(P) \subseteq \mathbf{fv}(Q)$ and $\mathbf{fv}(N) \subseteq \mathbf{fv}(M)$. Then $\mathbf{fv}(P \to N) = \mathbf{fv}(P) \cup \mathbf{fv}(N) \subseteq \mathbf{fv}(Q) \cup \mathbf{fv}(M) = \mathbf{fv}(Q \to M)$.

Case 4. $T \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant \forall\overrightarrow{\beta^+}.\, M$

$$\begin{aligned}
\mathbf{fv}\,\forall\overrightarrow{\alpha^+}.\, N &\subseteq \mathbf{fv}\,([\overrightarrow{P}/\overrightarrow{\alpha^+}]N) \setminus \overrightarrow{\beta^+} && \text{here } \overrightarrow{\beta^+} \text{ is excluded by the premise } \mathbf{fv}\,N \cap \overrightarrow{\beta^+} = \emptyset\\
&\subseteq \mathbf{fv}\,M \setminus \overrightarrow{\beta^+} && \text{by the induction hypothesis, } \mathbf{fv}\,([\overrightarrow{P}/\overrightarrow{\alpha^+}]N) \subseteq \mathbf{fv}\,M\\
&\subseteq \mathbf{fv}\,\forall\overrightarrow{\beta^+}.\, M
\end{aligned}$$

Case 5. The positive cases are symmetric.

□

Corollary 5 (Free Variables of mutual subtypes).

– *If* $T \vdash N \simeq^{\leqslant} M$ *then* $\mathbf{fv}\,N = \mathbf{fv}\,M$,
+ *If* $T \vdash P \simeq^{\leqslant} Q$ *then* $\mathbf{fv}\,P = \mathbf{fv}\,Q$

Corollary 6. *Suppose that all the types below are well-formed in T and $T' \subseteq T$. Then*

+ $T \vdash P \simeq^{\leqslant} Q$ *implies* $T' \vdash P \iff T' \vdash Q$
– $T \vdash N \simeq^{\leqslant} M$ *implies* $T' \vdash N \iff T' \vdash M$

Proof. From lemma 2 and corollary 5. □

Lemma 16 (Decomposition of quantifier rules). *Assuming that $\overrightarrow{\alpha^+}$, $\overrightarrow{\beta^+}$, $\overrightarrow{\alpha^-}$, and $\overrightarrow{\alpha^-}$ are disjoint from T,*

$-_R$ $T \vdash N \leqslant \forall\overrightarrow{\beta^+}.\, M$ *holds if and only if* $T, \overrightarrow{\beta^+} \vdash N \leqslant M$;
$+_R$ $T \vdash P \geqslant \exists\overrightarrow{\beta^-}.\, Q$ *holds if and only if* $T, \overrightarrow{\beta^-} \vdash P \geqslant Q$;
$-_L$ *suppose* $M \neq \forall\ldots$ *then* $T \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant M$ *holds if and only if* $T \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N \leqslant M$ *for some* $T \vdash \overrightarrow{P}$;
$+_L$ *suppose* $Q \neq \exists\ldots$ *then* $T \vdash \exists\overrightarrow{\alpha^-}.\, P \geqslant Q$ *holds if and only if* $T \vdash [\overrightarrow{N}/\overrightarrow{\alpha^-}]P \geqslant Q$ *for some* $T \vdash \overrightarrow{N}$.

Proof.

$-_R$ Let us prove both directions.
⇒ Let us assume $T \vdash N \leqslant \forall\overrightarrow{\beta^+}.\, M$. $T \vdash N \leqslant \forall\overrightarrow{\beta^+}.\, M$. Let us decompose M as $\forall\overrightarrow{\beta^+}'.\, M'$ where M' does not start with $\forall$, and decompose N as $\forall\overrightarrow{\alpha^+}.\, N'$ where N' does not start with $\forall$. If $\overrightarrow{\beta^+}$ is empty, then $T, \overrightarrow{\beta^+} \vdash N \leqslant M$ holds by assumption. Otherwise, $T \vdash \forall\overrightarrow{\alpha^+}.\, N' \leqslant \forall\overrightarrow{\beta^+}.\, \forall\overrightarrow{\beta^+}'.\, M$ is inferred by $(\forall^{\leqslant})$, and by inversion: $T, \overrightarrow{\beta^+}, \overrightarrow{\beta^+}' \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant$

M' for some $T, \overrightarrow{\beta^+}, \overrightarrow{\beta^{+\prime}} \vdash \overrightarrow{P}$. Then again by $(\forall^{\leqslant})$ with the same $\overrightarrow{P}$, $T, \overrightarrow{\beta^+} \vdash \forall\overrightarrow{\alpha^+}.\, N' \leqslant \forall\overrightarrow{\beta^{+\prime}}.\, M'$, that is $T, \overrightarrow{\beta^+} \vdash N \leqslant M$.

$\Leftarrow$ let us assume $T, \overrightarrow{\beta^+} \vdash N \leqslant M$, and let us decompose N as $\forall\overrightarrow{\alpha^+}.\, N'$ where N' does not start with $\forall$, and M as $\forall\overrightarrow{\beta^{+\prime}}.\, M'$ where M' does not start with $\forall$. if $\overrightarrow{\alpha^+}$ and $\overrightarrow{\beta^{+\prime}}$ are empty then $T, \overrightarrow{\beta^+} \vdash N \leqslant M$ is turned into $T \vdash N \leqslant \forall\overrightarrow{\beta^+}.\, M$ by $(\forall^{\leqslant})$. Otherwise, $T, \overrightarrow{\beta^+} \vdash \forall\overrightarrow{\alpha^+}.\, N' \leqslant \forall\overrightarrow{\beta^{+\prime}}.\, M'$ is inferred by $(\forall^{\leqslant})$, that is $T, \overrightarrow{\beta^+}, \overrightarrow{\beta^{+\prime}} \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant M'$ for some $T, \overrightarrow{\beta^+}, \overrightarrow{\beta^{+\prime}} \vdash \overrightarrow{P}$. Then by $(\forall^{\leqslant})$ again, $T \vdash \forall\overrightarrow{\alpha^+}.\, N' \leqslant \forall\overrightarrow{\beta^+}, \overrightarrow{\beta^{+\prime}}.\, M'$, in other words, $T \vdash \forall\overrightarrow{\alpha^+}.\, N' \leqslant \forall\overrightarrow{\beta^+}.\, \forall\overrightarrow{\beta^{+\prime}}.\, M'$, that is $T \vdash N \leqslant \forall\overrightarrow{\beta^+}.\, M$.

$-_L$ Suppose $M \neq \forall \ldots$. Let us prove both directions.

$\Rightarrow$ Let us assume $T \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant M$. then if $\overrightarrow{\alpha^+} = \cdot$, $T \vdash N \leqslant M$ holds immediately. Otherwise, let us decompose N as $\forall\overrightarrow{\alpha^{+\prime}}.\, N'$ where N' does not start with $\forall$. Then $T \vdash \forall\overrightarrow{\alpha^+}.\, \forall\overrightarrow{\alpha^{+\prime}}.\, N' \leqslant M'$ is inferred by $(\forall^{\leqslant})$, and by inversion, there exist $T \vdash \overrightarrow{P}, \overrightarrow{P}'$ such that $T \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}][\overrightarrow{P}'/\overrightarrow{\alpha^{+\prime}}]N' \leqslant M'$ (the decomposition of substitutions is possible since $\overrightarrow{\alpha^+} \cap T = \emptyset$). Then by $(\forall^{\leqslant})$ again, $T \vdash \forall\overrightarrow{\alpha^{+\prime}}.\, [\overrightarrow{P}'/\overrightarrow{\alpha^{+\prime}}]N' \leqslant M'$ (notice that $[\overrightarrow{P}'/\overrightarrow{\alpha^{+\prime}}]N'$ cannot start with $\forall$).

$\Leftarrow$ Let us assume $T \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N \leqslant M$ for some $T \vdash \overrightarrow{P}$. let us decompose N as $\forall\overrightarrow{\alpha^{+\prime}}.\, N'$ where N' does not start with $\forall$. Then $T \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]\forall\overrightarrow{\alpha^{+\prime}}.\, N' \leqslant M'$ or, equivalently, $T \vdash \forall\overrightarrow{\alpha^{+\prime}}.\, [\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant M'$ is inferred by $(\forall^{\leqslant})$ (notice that $[\overrightarrow{P}/\overrightarrow{\alpha^+}]N'$ cannot start with $\forall$). By inversion, there exist $T \vdash \overrightarrow{P}'$ such that $T \vdash [\overrightarrow{P}'/\overrightarrow{\alpha^{+\prime}}][\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant M'$. Since $\overrightarrow{\alpha^{+\prime}}$ is disjoint from the free variables of $\overrightarrow{P}$ and from $\overrightarrow{\alpha^+}$, the composition of $\overrightarrow{P}'/\overrightarrow{\alpha^{+\prime}}$ and $\overrightarrow{P}/\overrightarrow{\alpha^+}$ can be joined into a single substitution well-formed in T. Then by $(\forall^{\leqslant})$ again, $T \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant M$.

$+$ The positive cases are proved symmetrically.

□

Corollary 7 (Redundant quantifier elimination).

$-_L$ *Suppose that* $\overrightarrow{\alpha^+} \cap \mathbf{fv}\,(N) = \emptyset$ *then* $T \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant M$ *holds if and only if* $T \vdash N \leqslant M$;

$-_R$ *Suppose that* $\overrightarrow{\alpha^+} \cap \mathbf{fv}\,(M) = \emptyset$ *then* $T \vdash N \leqslant \forall\overrightarrow{\alpha^+}.\, M$ *holds if and only if* $T \vdash N \leqslant M$;

$+_L$ *Suppose that* $\overrightarrow{\alpha^-} \cap \mathbf{fv}\,(P) = \emptyset$ *then* $T \vdash \exists\overrightarrow{\alpha^-}.\, P \geqslant Q$ *holds if and only if* $T \vdash P \geqslant Q$.

$+_R$ *Suppose that* $\overrightarrow{\alpha^-} \cap \mathbf{fv}\,(Q) = \emptyset$ *then* $T \vdash P \geqslant \exists\overrightarrow{\alpha^-}.\, Q$ *holds if and only if* $T \vdash P \geqslant Q$.

Proof. $-_R$ Suppose that $\overrightarrow{\alpha^+} \cap \mathbf{fv}\,(M) = \emptyset$ then by lemma 16, $T \vdash N \leqslant \forall\overrightarrow{\alpha^+}.\, M$ is equivalent to $T, \overrightarrow{\alpha^+} \vdash N \leqslant M$, By lemma 2, since $\overrightarrow{\alpha^+} \cap \mathbf{fv}\,(N) = \emptyset$ and $\overrightarrow{\alpha^+} \cap \mathbf{fv}\,(M) = \emptyset$, $T, \overrightarrow{\alpha^+} \vdash N \leqslant M$ is equivalent to $T \vdash N \leqslant M$.

$-_L$ Suppose that $\overrightarrow{\alpha^+} \cap \mathbf{fv}\,(N) = \emptyset$. Let us decompose M as $\forall\overrightarrow{\beta^+}.\, M'$ where M' does not start with $\forall$. By lemma 16, $T \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant \forall\overrightarrow{\beta^+}.\, M'$ is equivalent to $T, \overrightarrow{\beta^+} \vdash \forall\overrightarrow{\alpha^+}.\, N \leqslant M'$, which is equivalent to existence of $T, \overrightarrow{\beta^+} \vdash \overrightarrow{P}$ such that $T, \overrightarrow{\beta^+} \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N \leqslant M'$. Since $[\overrightarrow{P}/\overrightarrow{\alpha^+}]N = N$, the latter is equivalent to $T, \overrightarrow{\beta^+} \vdash N \leqslant M'$, which is equivalent to $T \vdash N \leqslant \forall\overrightarrow{\beta^+}.\, M'$. $T, \overrightarrow{\beta^+} \vdash \overrightarrow{P}$ can be chosen arbitrary, for example, $\overrightarrow{P}_i = \exists\alpha^-.\, \downarrow\alpha^-$.

$+$ The positive cases are proved symmetrically.

□

Lemma 17 (Subtypes and supertypes of a variable). *Assuming* $T \vdash \alpha^-$, $T \vdash \alpha^+$, $T \vdash N$, *and* $T \vdash P$,

$+$ *if* $T \vdash P \geqslant \exists\overrightarrow{\alpha^-}.\, \alpha^+$ *or* $T \vdash \exists\overrightarrow{\alpha^-}.\, \alpha^+ \geqslant P$ *then* $P = \exists\overrightarrow{\beta^-}.\, \alpha^+$ *(for some potentially empty* $\overrightarrow{\beta^-}$*)*

$-$ *if* $T \vdash N \leqslant \forall\overrightarrow{\alpha^+}.\, \alpha^-$ *or* $T \vdash \forall\overrightarrow{\alpha^+}.\, \alpha^- \leqslant N$ *then* $N = \forall\overrightarrow{\beta^+}.\, \alpha^-$ *(for some potentially empty* $\overrightarrow{\beta^+}$*)*

Proof. We prove by induction on the tree inferring $T \vdash P \geqslant \exists\overrightarrow{\alpha^-}.\,\alpha^+$ or $T \vdash \exists\overrightarrow{\alpha^-}.\,\alpha^+ \geqslant P$ or or $T \vdash N \leqslant \forall\overrightarrow{\alpha^+}.\,\alpha^-$ or $T \vdash \forall\overrightarrow{\alpha^+}.\,\alpha^- \leqslant N$.

Let us consider which one of these judgments is inferred.

Case 1. $T \vdash P \geqslant \exists\overrightarrow{\alpha^-}.\,\alpha^+$

If the size of the inference tree is 1 then the only rule that can infer it is ($\text{Var}^{\geqslant}_{+}$), which implies that $\overrightarrow{\alpha^-}$ is empty and $P = \alpha^+$.

If the size of the inference tree is > 1 then the last rule inferring it must be ($\exists^{\geqslant}$). By inverting this rule, $P = \exists\overrightarrow{\beta^-}.\,P'$ where P' does not start with $\exists$ and $T, \overrightarrow{\alpha^-} \vdash [\overrightarrow{N}/\overrightarrow{\beta^-}]P' \geqslant \alpha^+$ for some $T, \overrightarrow{\alpha^-} \vdash N_i$.

By the induction hypothesis, $[\overrightarrow{N}/\overrightarrow{\beta^-}]P' = \exists\overrightarrow{\gamma^-}.\,\alpha^+$. What shape can P' have? As mentioned, it does not start with $\exists$, and it cannot start with $\uparrow$ (otherwise, $[\overrightarrow{N}/\overrightarrow{\alpha^-}]P'$ would also start with $\uparrow$ and would not be equal to $\exists\overrightarrow{\beta^-}.\,\alpha^+$). This way, P' is a *positive* variable. As such, $[\overrightarrow{N}/\overrightarrow{\alpha^-}]P' = P'$, and then $P' = \exists\overrightarrow{\gamma^-}.\,\alpha^+$ meaning that $\overrightarrow{\gamma^-}$ is empty and $P' = \alpha^+$. This way, $P = \exists\overrightarrow{\beta^-}.\,P' = \exists\overrightarrow{\beta^-}.\,\alpha^+$, as required.

Case 2. $T \vdash \exists\overrightarrow{\alpha^-}.\,\alpha^+ \geqslant P$

If the size of the inference tree is 1 then the only rule that can infer it is ($\text{Var}^{\geqslant}_{+}$), which implies that $\overrightarrow{\alpha^-}$ is empty and $P = \alpha^+$.

If the size of the inference tree is > 1 then the last rule inferring it must be ($\exists^{\geqslant}$). By inverting this rule, $P = \exists\overrightarrow{\beta^-}.\,Q$ where $T, \overrightarrow{\beta^-} \vdash [\overrightarrow{N}/\overrightarrow{\alpha^-}]\alpha^+ \geqslant Q$ and Q does not start with $\exists$. Notice that since α^+ is positive, $[\overrightarrow{N}/\overrightarrow{\alpha^-}]\alpha^+ = \alpha^+$, i.e. $T, \overrightarrow{\beta^-} \vdash \alpha^+ \geqslant Q$.

By the induction hypothesis, $Q = \exists\overrightarrow{\beta^-}'.\,\alpha^+$, and since Q does not start with $\exists$, $\overrightarrow{\beta^-}'$ is empty This way, $P = \exists\overrightarrow{\beta^-}.\,Q = \exists\overrightarrow{\beta^-}.\,\alpha^+$, as required.

Case 3. The negative cases ($T \vdash N \leqslant \forall\overrightarrow{\alpha^+}.\,\alpha^-$ and $T \vdash \forall\overrightarrow{\alpha^+}.\,\alpha^- \leqslant N$) are proved analogously.

□

Corollary 8 (Variables have no proper subtypes and supertypes). *Assuming that all mentioned types are well-formed in T,*

$$\begin{aligned}
T \vdash P \geqslant \alpha^+ &\iff P = \exists\overrightarrow{\beta^-}.\,\alpha^+ \iff T \vdash P \simeq^{\leqslant} \alpha^+ \iff P \simeq^{D} \alpha^+\\
T \vdash \alpha^+ \geqslant P &\iff P = \exists\overrightarrow{\beta^-}.\,\alpha^+ \iff T \vdash P \simeq^{\leqslant} \alpha^+ \iff P \simeq^{D} \alpha^+\\
T \vdash N \leqslant \alpha^- &\iff N = \forall\overrightarrow{\beta^+}.\,\alpha^- \iff T \vdash N \simeq^{\leqslant} \alpha^- \iff N \simeq^{D} \alpha^-\\
T \vdash \alpha^- \leqslant N &\iff N = \forall\overrightarrow{\beta^+}.\,\alpha^- \iff T \vdash N \simeq^{\leqslant} \alpha^- \iff N \simeq^{D} \alpha^-
\end{aligned}$$

Proof. Notice that $T \vdash \exists\overrightarrow{\beta^-}.\,\alpha^+ \simeq^{\leqslant} \alpha^+$ and $\exists\overrightarrow{\beta^-}.\,\alpha^+ \simeq^{D} \alpha^+$ and apply lemma 17. □

Lemma 18 (Subtyping context irrelevance). *Suppose that all the mentioned types are well-formed in T_1 and T_2. Then*

+ *$T_1 \vdash P \geqslant Q$ is equivalent to $T_2 \vdash P \geqslant Q$;*
− *$T_1 \vdash N \leqslant M$ is equivalent to $T_2 \vdash N \leqslant M$.*

Proof. We prove it by induction on the size of $T_1 \vdash P \geqslant Q$ and mutually, the size of $T_1 \vdash N \leqslant M$.

All the cases except ($\exists^{\geqslant}$) and ($\forall^{\leqslant}$) are proven congruently: first, we apply the inversion to $T_1 \vdash P \geqslant Q$ to obtain the premises of the corresponding rule X, then we apply the induction hypothesis to each premise, and build the inference tree (with T_2) by the same rule X.

Suppose that the judgment is inferred by ($\exists^{\geqslant}$). Then we are proving that $T_1 \vdash \exists\overrightarrow{\alpha^-}.\,P \geqslant \exists\overrightarrow{\beta^-}.\,Q$ implies $T_2 \vdash \exists\overrightarrow{\alpha^-}.\,P \geqslant \exists\overrightarrow{\beta^-}.\,Q$ (the other implication is proven symmetrically).

By inversion of $T_1 \vdash \exists\overrightarrow{\alpha^-}.\,P \geqslant \exists\overrightarrow{\beta^-}.\,Q$, we obtain σ such that $T_1, \overrightarrow{\beta^-} \vdash \sigma : \overrightarrow{\alpha^-}$ and $T_1, \overrightarrow{\beta^-} \vdash [\sigma]P \geqslant Q$. By lemma 15, $\mathbf{fv}\,([\sigma]P) \subseteq \mathbf{fv}\,(Q)$.

From the well-formedness statements $T_i \vdash \exists\overrightarrow{\alpha^-}.\,P$ and $T_i \vdash \exists\overrightarrow{\beta^-}.\,Q$ we have:

- $T_1, \overrightarrow{\alpha^-} \vdash P$, which also means $T_1, \overrightarrow{\beta^-} \vdash [\sigma]P$ by lemma 3;
- $T_2, \overrightarrow{\alpha^-} \vdash P$;
- $T_1, \overrightarrow{\beta^-} \vdash Q$; and
- $T_2, \overrightarrow{\beta^-} \vdash Q$, which means $\mathbf{fv}\,(Q) \subseteq T_2, \overrightarrow{\beta^-}$ by lemma 1, and combining it with $\mathbf{fv}\,([\sigma]P) \subseteq \mathbf{fv}\,(Q)$, we have $\mathbf{fv}\,([\sigma]P) \subseteq T_2, \overrightarrow{\beta^-}$.

Let us construct a substitution σ_0 in the following way:

$$\begin{cases} [\sigma_0]\alpha_i^- = [\sigma]\alpha_i^- & \text{for } \alpha_i^- \in \overrightarrow{\alpha^-} \cap \mathbf{fv}\,(P) \\ [\sigma_0]\alpha_i^- = \forall\gamma^+.\,\uparrow\gamma^+ & \text{for } \alpha_i^- \in \overrightarrow{\alpha^-} \setminus \mathbf{fv}\,(P) \\ [\sigma_0]\gamma^\pm = \gamma^\pm & \text{for any other } \gamma^\pm \end{cases}$$

Notice that

(1) $[\sigma_0]P = [\sigma]P$. Since $\sigma_0|_{\mathbf{fv}\,(P)} = \sigma|_{\mathbf{fv}\,(P)}$ as functions (which follows from the construction of σ_0 and the signature of σ), $[\sigma_0]P = [\sigma_0|_{\mathbf{fv}\,(P)}]P = [\sigma|_{\mathbf{fv}\,(P)}]P = [\sigma]P$ (where the first and the last equalities are by lemma 4).

(2) $\mathbf{fv}\,([\sigma]P) \vdash \sigma_0 : \overrightarrow{\alpha^-}$. To show that, let us consider α_i^-

- if $\alpha_i^- \in \overrightarrow{\alpha^-} \setminus \mathbf{fv}\,(P)$ then $\cdot \vdash [\sigma_0]\alpha_i^-$, which can be weakened to $\mathbf{fv}\,([\sigma]P) \vdash [\sigma_0]\alpha_i^-$;
- if $\alpha_i^- \in \overrightarrow{\alpha^-} \cap \mathbf{fv}\,(P)$, we have $[\sigma_0]\alpha_i^- = [\sigma]\alpha_i^-$, and thus, by specification of σ, $T_1, \overrightarrow{\beta^+} \vdash [\sigma_0]\alpha_i^-$. By corollary 1, it means $\mathbf{fv}\,([\sigma_0]\alpha_i^-) \vdash [\sigma_0]\alpha_i^-$, which we weaken (corollary 2) to $\mathbf{fv}\,([\sigma]P) \vdash [\sigma_0]\alpha_i^-$ (since $\mathbf{fv}\,([\sigma_0]\alpha_i^-) \subseteq \mathbf{fv}\,([\sigma_0]P)$ by lemma 14, and $[\sigma_0]P = [\sigma]P$, as noted above).

By corollary 1, $T_1, \overrightarrow{\beta^-} \vdash [\sigma]P$ implies $\mathbf{fv}\,([\sigma]P) \vdash [\sigma]P$, which, since $\mathbf{fv}\,([\sigma]P) \subseteq T_2, \overrightarrow{\beta^-}$, is weakened to $T_2, \overrightarrow{\beta^-} \vdash [\sigma]P$. and rewritten as $T_2, \overrightarrow{\beta^-} \vdash [\sigma_0]P$.

Notice that the premises of the induction hold:

(1) $T_i, \overrightarrow{\beta^-} \vdash [\sigma_0]P$,
(2) $T_i, \overrightarrow{\beta^-} \vdash Q$, and
(3) $T_1, \overrightarrow{\beta^-} \vdash [\sigma_0]P \geqslant Q$, notice that the tree inferring this judgment is the same tree inferring $T_1, \overrightarrow{\beta^-} \vdash [\sigma]P \geqslant Q$ (since $[\sigma_0]P = [\sigma]P$), i.e., it is a subtree of $T_1 \vdash \exists\overrightarrow{\alpha^-}.\,P \geqslant \exists\overrightarrow{\beta^-}.\,Q$.

This way, by the induction hypothesis, $T_2, \overrightarrow{\beta^-} \vdash [\sigma_0]P \geqslant Q$. Combining it with $T_2, \overrightarrow{\beta^-} \vdash \sigma_0 : \overrightarrow{\alpha^-}$ by ($\exists^{\geqslant}$), we obtain $T_2 \vdash \exists\overrightarrow{\alpha^-}.\,P \geqslant \exists\overrightarrow{\beta^-}.\,Q$.

The case of $T_1 \vdash \forall\overrightarrow{\alpha^+}.\,N \leqslant \forall\overrightarrow{\beta^+}.\,M$ is symmetric. □

Lemma 19 (Weakening of subtyping context). *Suppose T_1 and T_2 are contexts and $T_1 \subseteq T_2$. Then*

+ *$T_1 \vdash P \geqslant Q$ implies $T_2 \vdash P \geqslant Q$;*
− *$T_1 \vdash N \leqslant M$ implies $T_2 \vdash N \leqslant M$.*

Proof. By straightforward induction on the subtyping derivation. The polymorphic cases follow from lemma 7. □

Lemma 20 (Reflexivity of subtyping). *Assuming all the types are well-formed in T,*

− $T \vdash N \leqslant N$
+ $T \vdash P \geqslant P$

Proof. Let us prove it by the size of N and mutually, P.

Case 1. $N = \alpha^-$

Then $T \vdash \alpha^- \leqslant \alpha^-$ is inferred immediately by (Var$^{\leqslant}_{-}$).

Case 2. $N = \forall\overrightarrow{\alpha^+}.\ N'$ where $\overrightarrow{\alpha^+}$ is not empty

First, we rename $\overrightarrow{\alpha^+}$ to fresh $\overrightarrow{\beta^+}$ in $\forall\overrightarrow{\alpha^+}.\ N'$ to avoid name clashes: $\forall\overrightarrow{\alpha^+}.\ N' = \forall\overrightarrow{\beta^+}.\ [\overrightarrow{\alpha^+}/\overrightarrow{\beta^+}]N'$. Then to infer $T \vdash \forall\overrightarrow{\alpha^+}.\ N' \leqslant \forall\overrightarrow{\beta^+}.\ [\overrightarrow{\alpha^+}/\overrightarrow{\beta^+}]N'$ we can apply ($\forall^{\leqslant}$), instantiating $\overrightarrow{\alpha^+}$ with $\overrightarrow{\beta^+}$:

- $\mathbf{fv}\,N \cap \overrightarrow{\beta^+} = \emptyset$ by choice of $\overrightarrow{\beta^+}$,
- $T, \overrightarrow{\beta^+} \vdash {\beta^+}_i$,
- $T, \overrightarrow{\beta^+} \vdash [\overrightarrow{\beta^+}/\overrightarrow{\alpha^+}]N' \leqslant [\overrightarrow{\beta^+}/\overrightarrow{\alpha^+}]N'$ by the induction hypothesis, since the size of $[\overrightarrow{\beta^+}/\overrightarrow{\alpha^+}]N'$ is equal to the size of N', which is smaller than the size of $N = \forall\overrightarrow{\alpha^+}.\ N'$.

Case 3. $N = P \to M$

Then $T \vdash P \to M \leqslant P \to M$ is inferred by ($\to^{\leqslant}$), since $T \vdash P \geqslant P$ and $T \vdash M \leqslant M$ hold the induction hypothesis.

Case 4. $N = \uparrow P$

Then $T \vdash \uparrow P \leqslant \uparrow P$ is inferred by ($\uparrow^{\leqslant}$), since $T \vdash P \geqslant P$ holds by the induction hypothesis.

Case 5. The positive cases are symmetric to the negative ones.

□

Lemma 21 (Substitution preserves subtyipng). *Suppose that all mentioned types are well-formed in T_1, and σ is a substitution $T_2 \vdash \sigma : T_1$.*

– *If $T_1 \vdash N \leqslant M$ then $T_2 \vdash [\sigma]N \leqslant [\sigma]M$.*
+ *If $T_1 \vdash P \geqslant Q$ then $T_2 \vdash [\sigma]P \geqslant [\sigma]Q$.*

Proof. We prove it by induction on the size of the derivation of $T_1 \vdash N \leqslant M$ and mutually, $T_1 \vdash P \geqslant Q$. Let us consider the last rule used in the derivation:

Case 1. (Var$^{\leqslant}_{-}$). Then by inversion, $N = \alpha^-$ and $M = \alpha^-$. By reflexivity of subtyping (lemma 20), we have $T_2 \vdash [\sigma]\alpha^- \leqslant [\sigma]\alpha^-$, i.e. $T_2 \vdash [\sigma]N \leqslant [\sigma]M$, as required.

Case 2. ($\forall^{\leqslant}$). Then by inversion, $N = \forall\overrightarrow{\alpha^+}.\ N'$, $M = \forall\overrightarrow{\beta^+}.\ M'$, where $\overrightarrow{\alpha^+}$ or $\overrightarrow{\beta^+}$ is not empty. Moreover, $T_1, \overrightarrow{\beta^+} \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant M'$ for some $T_1, \overrightarrow{\beta^+} \vdash \overrightarrow{P}$, and $\mathbf{fv}\,N \cap \overrightarrow{\beta^+} = \emptyset$.

Notice that since the derivation of $T_1, \overrightarrow{\beta^+} \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant M'$ is a subderivation of the derivation of $T \vdash N \leqslant M$, its size is smaller, and hence, the induction hypothesis applies ($T_1, \overrightarrow{\beta^+} \vdash \sigma : T_1, \overrightarrow{\beta^+}$ by lemma 12) : $T_2, \overrightarrow{\beta^+} \vdash [\sigma][\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant [\sigma]M'$.

Notice that by convention, $\overrightarrow{\alpha^+}$ and $\overrightarrow{\beta^+}$ are fresh, and thus, $[\sigma]\forall\overrightarrow{\alpha^+}.\ N' = \forall\overrightarrow{\alpha^+}.\ [\sigma]N'$ and $[\sigma]\forall\overrightarrow{\beta^+}.\ M' = \forall\overrightarrow{\beta^+}.\ [\sigma]M'$, which means that the required $T_2, T \vdash [\sigma]\forall\overrightarrow{\alpha^+}.\ N' \leqslant [\sigma]\forall\overrightarrow{\beta^+}.\ M'$ is rewritten as $T_2, T \vdash \forall\overrightarrow{\alpha^+}.\ [\sigma]N' \leqslant \forall\overrightarrow{\beta^+}.\ [\sigma]M'$.

To infer it, we apply ($\forall^{\leqslant}$), instantiating ${\alpha^+}_i$ with $[\sigma]P_i$:

- $\mathbf{fv}\,[\sigma]N \cap \overrightarrow{\beta^+} = \emptyset$;
- $T_2, T, \overrightarrow{\beta^+} \vdash [\sigma]P_i$, by lemma 3 since from the inversion, $T_1, T, \overrightarrow{\beta^+} \vdash P_i$;
- $T, \overrightarrow{\beta^+} \vdash [[\sigma]\overrightarrow{P}/\overrightarrow{\alpha^+}][\sigma]N' \leqslant [\sigma]M'$ holds by lemma 11: Since $\overrightarrow{\alpha^+}$ is fresh, it is disjoint with the domain and the codomain of σ (T_1 and T_2), and thus, $[\sigma][\overrightarrow{P}/\overrightarrow{\alpha^+}]N' = [\sigma \ll \overrightarrow{P}/\overrightarrow{\alpha^+}][\sigma]N' = [[\sigma]\overrightarrow{P}/\overrightarrow{\alpha^+}][\sigma]N'$. Then $T_2, T, \overrightarrow{\beta^+} \vdash [\sigma][\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant [\sigma]M'$ holds by the induction hypothesis.

Case 3. ($\to^{\leqslant}$). Then by inversion, $N = P \to N_1$, $M = Q \to M_1$, $T \vdash P \geqslant Q$, and $T \vdash N_1 \leqslant M_1$. And by the induction hypothesis, $T' \vdash [\sigma]P \geqslant [\sigma]Q$ and $T' \vdash [\sigma]N_1 \leqslant [\sigma]M_1$. Then $T' \vdash [\sigma]N \leqslant [\sigma]M$, i.e. $T' \vdash [\sigma]P \to [\sigma]N_1 \leqslant [\sigma]Q \to [\sigma]M_1$, is inferred by ($\to^{\leqslant}$).

Case 4. ($\uparrow^{\leqslant}$). Then by inversion, $N = \uparrow P$, $M = \uparrow Q$, and $T \vdash P \simeq^{\leqslant} Q$, which by inversion means that $T \vdash P \geqslant Q$ and $T \vdash Q \geqslant P$. Then the induction hypothesis applies, and we have

$T' \vdash [\sigma]P \geqslant [\sigma]Q$ and $T' \vdash [\sigma]Q \geqslant [\sigma]P$. Then by sequential application of ($\simeq^{\leqslant}_{-}$) and ($\uparrow^{\leqslant}$) to these judgments, we have $T' \vdash \uparrow[\sigma]P \leqslant \uparrow[\sigma]Q$, i.e. $T' \vdash [\sigma]N \leqslant [\sigma]M$, as required.

Case 5. The positive cases are proved symmetrically.

□

Corollary 9 (Substitution preserves subtyping induced equivalence). *Suppose that $T \vdash \sigma : T_1$. Then*

+ *if $T_1 \vdash P$, $T_1 \vdash Q$, and $T_1 \vdash P \simeq^{\leqslant} Q$ then $T \vdash [\sigma]P \simeq^{\leqslant} [\sigma]Q$*
− *if $T_1 \vdash N$, $T_1 \vdash M$, and $T_1 \vdash N \simeq^{\leqslant} M$ then $T \vdash [\sigma]N \simeq^{\leqslant} [\sigma]M$*

Lemma 22 (Transitivity of subtyping). *Assuming the types are well-formed in T,*

− *if $T \vdash N_1 \leqslant N_2$ and $T \vdash N_2 \leqslant N_3$ then $T \vdash N_1 \leqslant N_3$,*
+ *if $T \vdash P_1 \geqslant P_2$ and $T \vdash P_2 \geqslant P_3$ then $T \vdash P_1 \geqslant P_3$.*

Proof. To prove it, we formulate a stronger property, which will imply the required one, taking $\sigma = T \vdash \mathrm{id} : T$.

Assuming all the types are well-formed in T,

− if $T \vdash N \leqslant M_1$, $T \vdash M_2 \leqslant K$, and for $T' \vdash \sigma : T$, $[\sigma]M_1 = [\sigma]M_2$ then $T' \vdash [\sigma]N \leqslant [\sigma]K$
+ if $T \vdash P \geqslant Q_1$, $T \vdash Q_2 \geqslant R$, and for $T' \vdash \sigma : T$, $[\sigma]Q_1 = [\sigma]Q_2$ then $T' \vdash [\sigma]P \geqslant [\sigma]R$

We prove it by induction on $\mathrm{size}(T \vdash N \leqslant M_1) + \mathrm{size}(T \vdash M_2 \leqslant K)$ and mutually, on $\mathrm{size}(T \vdash P \geqslant Q_1) + \mathrm{size}(T \vdash Q_2 \geqslant R)$.

First, let us consider the 3 important cases.

Case 1. Let us consider the case when $M_1 = \forall\overrightarrow{\beta^+}_1.\ \alpha^-$. Then by lemma 17, $T \vdash N \leqslant M_1$ means that $N = \forall\overrightarrow{\alpha^+}.\ \alpha^-$. $[\sigma]M_1 = [\sigma]M_2$ means that $\forall\overrightarrow{\beta^+}_1.\ [\sigma]\alpha^- = [\sigma]M_2$. Applying σ to both sides of $T \vdash M_2 \leqslant K$ (by lemma 21), we obtain $T' \vdash [\sigma]M_2 \leqslant [\sigma]K$, that is $T' \vdash \forall\overrightarrow{\beta^+}_1.\ [\sigma]\alpha^- \leqslant [\sigma]K$. Since $\mathbf{fv}\,([\sigma]\alpha^-) \subseteq T, \alpha^-$, it is disjoint from $\overrightarrow{\alpha^+}$ and $\overrightarrow{\beta^+}_1$, This way, by corollary 7, $T' \vdash \forall\overrightarrow{\beta^+}_1.\ [\sigma]\alpha^- \leqslant [\sigma]K$ is equivalent to $T' \vdash [\sigma]\alpha^- \leqslant [\sigma]K$, which is equivalent to $T' \vdash \forall\overrightarrow{\alpha^+}.\ [\sigma]\alpha^- \leqslant [\sigma]K$, that is $T' \vdash [\sigma]N \leqslant [\sigma]K$.

Case 2. Let us consider the case when $M_2 = \forall\overrightarrow{\beta^+}_2.\ \alpha^-$. This case is symmetric to the previous one. Notice that lemma 17 and corollary 7 are agnostic to the side on which the quantifiers occur, and thus, the proof stays the same.

Case 3. Let us decompose the types, by extracting the outer quantifiers:

- $N = \forall\overrightarrow{\alpha^+}.\ N'$, where $N' \neq \forall\ldots$,
- $M_1 = \forall\overrightarrow{\beta^+}_1.\ M'_1$, where $M'_1 \neq \forall\ldots$,
- $M_2 = \forall\overrightarrow{\beta^+}_2.\ M'_2$, where $M'_2 \neq \forall\ldots$,
- $K = \forall\overrightarrow{\gamma^+}.\ K'$, where $K' \neq \forall\ldots$.

and assume that at least one of $\overrightarrow{\alpha^+}$, $\overrightarrow{\beta^+}_1$, $\overrightarrow{\beta^+}_2$, and $\overrightarrow{\gamma^+}$ is not empty. Since $[\sigma]M_1 = [\sigma]M_2$, we have $\forall\overrightarrow{\beta^+}_1.\ [\sigma]M'_1 = \forall\overrightarrow{\beta^+}_2.\ [\sigma]M'_2$, and since M'_i are not variables (which was covered by the previous cases) and do not start with $\forall$, $[\sigma]M'_i$ do not start with $\forall$ either, which means $\overrightarrow{\beta^+}_1 = \overrightarrow{\beta^+}_2$ and $[\sigma]M'_1 = [\sigma]M'_2$. Let us rename $\overrightarrow{\beta^+}_1$ and $\overrightarrow{\beta^+}_2$ to $\overrightarrow{\beta^+}$. Then $M_1 = \forall\overrightarrow{\beta^+}.\ M'_1$ and $M_2 = \forall\overrightarrow{\beta^+}.\ M'_2$.

By lemma 16 applied twice to $T \vdash \forall\overrightarrow{\alpha^+}.\ N' \leqslant \forall\overrightarrow{\beta^+}.\ M'_1$ and to $T \vdash \forall\overrightarrow{\beta^+}.\ M'_2 \leqslant \forall\overrightarrow{\gamma^+}.\ K'$, we have the following:

(1) $T, \overrightarrow{\beta^+} \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant M'_1$ for some $T, \overrightarrow{\beta^+} \vdash \overrightarrow{P}$;
(2) $T, \overrightarrow{\gamma^+} \vdash [\overrightarrow{Q}/\overrightarrow{\beta^+}]M'_2 \leqslant K'$ for some $T, \overrightarrow{\gamma^+} \vdash \overrightarrow{Q}$.

And since at least one of $\overrightarrow{\alpha^+}$, $\overrightarrow{\beta^+}$, and $\overrightarrow{\gamma^+}$ is not empty, either $T \vdash N \leqslant M_1$ or $T \vdash M_2 \leqslant K$ is inferred by ($\forall^{\leqslant}$), meaning that either $T, \overrightarrow{\beta^+} \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant M_1'$ is a proper subderivation of $T \vdash N \leqslant M_1$ or $T, \overrightarrow{\gamma^+} \vdash [\overrightarrow{Q}/\overrightarrow{\beta^+}]M_2' \leqslant K'$ is a proper subderivation of $T \vdash M_2 \leqslant K$.
Notice that we can weaken and rearrange the contexts without changing the sizes of the derivations: $T, \overrightarrow{\beta^+}, \overrightarrow{\gamma^+} \vdash [\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant M_1'$ and $T, \overrightarrow{\beta^+}, \overrightarrow{\gamma^+} \vdash [\overrightarrow{Q}/\overrightarrow{\beta^+}]M_2' \leqslant K'$. This way, the sum of the sizes of these derivations is smaller than the sum of the sizes of $T \vdash N \leqslant M_1$ and $T \vdash M_2 \leqslant K$. Let us apply the induction hypothesis to these derivations, with the substitution $T', \overrightarrow{\gamma^+} \vdash \sigma \circ (\overrightarrow{Q}/\overrightarrow{\beta^+}) : T, \overrightarrow{\beta^+}, \overrightarrow{\gamma^+}$ (lemma 12). To apply the induction hypothesis, it is left to show that $\sigma \circ (\overrightarrow{Q}/\overrightarrow{\beta^+})$ unifies M_1' and $[\overrightarrow{Q}/\overrightarrow{\beta^+}]M_2'$:

$$\begin{aligned}
[\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+}]M_1' &= [\sigma][\overrightarrow{Q}/\overrightarrow{\beta^+}]M_1' \\
&= [[\sigma]\overrightarrow{Q}/\overrightarrow{\beta^+}][\sigma]M_2' && \text{by lemma 11} \\
&= [[\sigma]\overrightarrow{Q}/\overrightarrow{\beta^+}][\sigma]M_2' && \text{Since } [\sigma]M_1' = [\sigma]M_2' \\
&= [\sigma][\overrightarrow{Q}/\overrightarrow{\beta^+}]M_2' && \text{by lemma 11} \\
&= [\sigma][\overrightarrow{Q}/\overrightarrow{\beta^+}][\overrightarrow{Q}/\overrightarrow{\beta^+}]M_2' && \text{Since } T, \overrightarrow{\gamma^+} \vdash \overrightarrow{Q}, \text{ and } (T, \overrightarrow{\gamma^+}) \cap \overrightarrow{\beta^+} = \emptyset \\
&= [\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+}][\overrightarrow{Q}/\overrightarrow{\beta^+}]M_2'
\end{aligned}$$

This way the induction hypothesis gives us $T', \overrightarrow{\gamma^+} \vdash [\sigma][\overrightarrow{Q}/\overrightarrow{\beta^+}][\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant [\sigma][\overrightarrow{Q}/\overrightarrow{\beta^+}]K'$, and since $T, \overrightarrow{\gamma^+} \vdash K'$, $[\overrightarrow{Q}/\overrightarrow{\beta^+}]K' = K'$, that is $T', \overrightarrow{\gamma^+} \vdash [\sigma][\overrightarrow{Q}/\overrightarrow{\beta^+}][\overrightarrow{P}/\overrightarrow{\alpha^+}]N' \leqslant [\sigma]K'$. Let us rewrite the substitution that we apply to N':

$$\begin{aligned}
[\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+} \circ \overrightarrow{P}/\overrightarrow{\alpha^+}]N' &= [(\sigma \lll \overrightarrow{Q}/\overrightarrow{\beta^+}) \circ \sigma \circ \overrightarrow{P}/\overrightarrow{\alpha^+}]N' && \text{by lemma 11} \\
&= [(\sigma \lll \overrightarrow{Q}/\overrightarrow{\beta^+}) \circ (\sigma \lll \overrightarrow{P}/\overrightarrow{\alpha^+}) \circ \sigma]N' && \text{by lemma 11} \\
&= [(((\sigma \lll \overrightarrow{Q}/\overrightarrow{\beta^+}) \circ \sigma) \lll \overrightarrow{P}/\overrightarrow{\alpha^+}) \circ \sigma]N' && \text{Since } \mathbf{fv}\,([\sigma]N') \cap \overrightarrow{\beta^+} = \emptyset \\
&= [((\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+}) \lll \overrightarrow{P}/\overrightarrow{\alpha^+}) \circ \sigma]N' && \text{by lemma 11} \\
&= [(\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+}) \lll \overrightarrow{P}/\overrightarrow{\alpha^+}][\sigma]N'
\end{aligned}$$

Notice that $(\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+}) \lll \overrightarrow{P}/\overrightarrow{\alpha^+}$ is a substitution that turns ${\alpha^+}_i$ into $[\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+}]P_i$, where $T', \overrightarrow{\gamma^+} \vdash [\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+}]P_i$. This way, $T', \overrightarrow{\gamma^+} \vdash [(\sigma \circ \overrightarrow{Q}/\overrightarrow{\beta^+}) \lll \overrightarrow{P}/\overrightarrow{\alpha^+}][\sigma]N' \leqslant [\sigma]K'$ means $T \vdash \forall\overrightarrow{\alpha^+}.\, [\sigma]N' \leqslant \forall\overrightarrow{\gamma^+}.\, [\sigma]K'$ by lemma 16, that is $T \vdash [\sigma]N \leqslant [\sigma]K$, as required.

Now, we can assume that neither $T \vdash N \leqslant M_1$ nor $T \vdash M_2 \leqslant K$ is inferred by ($\forall^{\leqslant}$), and that neither M_1 nor M_2 is equivalent to a variable. Because of that, $[\sigma]M_1 = [\sigma]M_2$ means that M_1 and M_2 have the same outer constructor. Let us consider the shape of M_1.

Case 1. $M_1 = \alpha^-$ this case has been considered;
Case 2. $M_1 = \forall\overrightarrow{\beta^+}.\, M_1'$ this case has been considered;
Case 3. $M_1 = \uparrow Q_1$. Then as noted above, $[\sigma]M_1 = [\sigma]M_2$ means that $M_2 = \uparrow Q_2$ and $[\sigma]Q_1 = [\sigma]Q_2$. Moreover, $T \vdash N \leqslant \uparrow Q_1$ can only be inferred by ($\uparrow^{\leqslant}$), and thus, $N = \uparrow P$, and by inversion, $T \vdash P \geqslant Q_1$ and $T \vdash Q_1 \geqslant P$. Analogously, $T \vdash \uparrow Q_2 \leqslant K$ means that $K = \uparrow R$, $T \vdash Q_2 \geqslant R$, and $T \vdash R \geqslant Q_2$.
Notice that the derivations of $T \vdash P \geqslant Q_1$ and $T \vdash Q_1 \geqslant P$ are proper sub-derivations of $T \vdash N \leqslant M_1$, and the derivations of $T \vdash Q_2 \geqslant R$ and $T \vdash R \geqslant Q_2$ are proper sub-derivations of $T \vdash M_2 \leqslant K$. This way, the induction hypothesis is applicable:

- applying the induction hypothesis to $T \vdash P \geqslant Q_1$ and $T \vdash Q_2 \geqslant R$ with $T' \vdash \sigma : T$ unifying Q_1 and Q_2, we obtain $T' \vdash [\sigma]P \geqslant [\sigma]R$;
- applying the induction hypothesis to $T \vdash R \geqslant Q_2$ and $T \vdash Q_1 \geqslant P$ with $T' \vdash \sigma : T$ unifying Q_2 and Q_1, we obtain $T' \vdash [\sigma]R \geqslant [\sigma]P$.

This way, by $(\uparrow^{\leqslant})$, $T' \vdash [\sigma]N \leqslant [\sigma]K$, as required.

Case 4. $M_1 = Q_1 \to M_1'$. Then as noted above, $[\sigma]M_1 = [\sigma]M_2$ means that $M_2 = Q_2 \to M_2'$, $[\sigma]Q_1 = [\sigma]Q_2$, and $[\sigma]M_1' = [\sigma]M_2'$. Moreover, $T \vdash N \leqslant Q_1 \to M_1'$ can only be inferred by $(\to^{\leqslant})$, and thus, $N = P \to N'$, and by inversion, $T \vdash P \geqslant Q_1$ and $T \vdash N' \leqslant M_1'$. Analogously, $T \vdash Q_2 \to M_2' \leqslant K$ means that $K = R \to K'$, $T \vdash Q_2 \geqslant R$, and $T \vdash M_2' \leqslant K'$.
Notice that the derivations of $T \vdash P \geqslant Q_1$ and $T \vdash N' \leqslant M_1'$ are proper sub-derivations of $T \vdash P \to N' \leqslant Q_1 \to M_1'$, and the derivations of $T \vdash Q_2 \geqslant R$ and $T \vdash M_2' \leqslant K'$ are proper sub-derivations of $T \vdash Q_2 \to M_2' \leqslant R \to K'$. This way, the induction hypothesis is applicable:

- applying the induction hypothesis to $T \vdash P \geqslant Q_1$ and $T \vdash Q_2 \geqslant R$ with $T' \vdash \sigma : T$ unifying Q_1 and Q_2, we obtain $T' \vdash [\sigma]P \geqslant [\sigma]R$;
- applying the induction hypothesis to $T \vdash N' \leqslant M_1'$ and $T \vdash M_2' \leqslant K'$ with $T' \vdash \sigma : T$ unifying M_1' and M_2', we obtain $T' \vdash [\sigma]N' \leqslant [\sigma]K'$.

This way, by $(\to^{\leqslant})$, $T' \vdash [\sigma]P \to [\sigma]N' \leqslant [\sigma]R \to [\sigma]K'$, that is $T' \vdash [\sigma]N \leqslant [\sigma]K$, as required.

After that, we consider all the analogous positive cases and prove them symmetrically. □

Corollary 10 (Transitivity of equivalence). *Assuming the types are well-formed in T,*

− *if $T \vdash N_1 \simeq^{\leqslant} N_2$ and $T \vdash N_2 \simeq^{\leqslant} N_3$ then $T \vdash N_1 \simeq^{\leqslant} N_3$,*
+ *if $T \vdash P_1 \simeq^{\leqslant} P_2$ and $T \vdash P_2 \simeq^{\leqslant} P_3$ then $T \vdash P_1 \simeq^{\leqslant} P_3$.*

8.4 Equivalence

Lemma 23 (Declarative Equivalence is invariant under bijections). *Suppose μ is a bijection $\mu : vars_1 \leftrightarrow vars_2$, then*

+ *$P_1 \simeq^D P_2$ implies $[\mu]P_1 \simeq^D [\mu]P_2$, and there exists an inference tree of $[\mu]P_1 \simeq^D [\mu]P_2$ with the same shape as the one inferring $P_1 \simeq^D P_2$;*
− *$N_1 \simeq^D N_2$ implies $[\mu]N_1 \simeq^D [\mu]N_2$, and there exists an inference tree of $[\mu]N_1 \simeq^D [\mu]N_2$ with the same shape as the one inferring $N_1 \simeq^D N_2$.*

Proof. We prove it by induction on $P_1 \simeq^D P_2$ and mutually, on $N_1 \simeq^D N_2$. Let us consider the last rule used in the derivation.

Case 1. $(\forall^{\simeq^D})$
Then we decompose N_1 as $\forall\overrightarrow{\alpha^+}_1.\, M_1$ and N_2 as $\forall\overrightarrow{\alpha^+}_2.\, M_2$, where M_1 and M_2 do not start with $\forall$-quantifiers. where $|\overrightarrow{\alpha^+}_1| + |\overrightarrow{\alpha^+}_2| > 0$. By convention, let us assume that $\overrightarrow{\alpha^+}_1$ and $\overrightarrow{\alpha^+}_2$ are disjoint form $vars_2$ and $vars_1$.
By inversion, $\overrightarrow{\alpha^+}_1 \cap \mathbf{fv}\, M_2 = \emptyset$ and $M_1 \simeq^D [\mu']M_2$ for some bijection $\mu' : (\overrightarrow{\alpha^+}_2 \cap \mathbf{fv}\, M_2) \leftrightarrow (\overrightarrow{\alpha^+}_1 \cap \mathbf{fv}\, M_1)$. Then let us apply the induction hypothesis to $M_1 \simeq^D [\mu']M_2$ to obtain $[\mu]M_1 \simeq^D [\mu][\mu']M_2$ inferred by the tree of the same shape as $M_1 \simeq^D [\mu']M_2$.
Notice that $[\mu]M_1$ and $[\mu]M_2$ do not start with $\forall$, That is $[\mu]\forall\overrightarrow{\alpha^+}_1.\, M_1 \simeq^D [\mu]\forall\overrightarrow{\alpha^+}_2.\, M_2$, rewritten as $\forall\overrightarrow{\alpha^+}_1.\, [\mu]M_1 \simeq^D \forall\overrightarrow{\alpha^+}_2.\, [\mu]M_2$, can be inferred by $(\forall^{\simeq^D})$:
(1) $\overrightarrow{\alpha^+}_1$ is disjoint from $vars_2 \cup \mathbf{fv}\, M_2 \subseteq \mathbf{fv}\, [\mu]M_2$;

(2) $[\mu]M_1 \simeq^D [\mu'][\mu]M_2$ because $[\mu'][\mu]M_2 = [\mu][\mu']M_2$ (by corollary 4: $\mu' : (\overrightarrow{\alpha^+}_2 \cap \mathbf{fv}\, M_2) \leftrightarrow (\overrightarrow{\alpha^+}_1 \cap \mathbf{fv}\, M_1)$, $\mu : vars_1 \leftrightarrow vars_2$, $vars_1$ is disjoint from $\overrightarrow{\alpha^+}_2$ and $\overrightarrow{\alpha^+}_1$; $\overrightarrow{\alpha^+}_2$ is disjoint from $vars_1$ and $vars_2$)

Notice that it is the same rule as the one inferring $N_1 \simeq^D N_2$, and thus, the shapes of the trees are the same.

Case 2. ($\text{VAR}_-^{\simeq^D}$)

Then $N_1 = N_2 = \alpha^-$, and the required $[\mu]\alpha^- = [\mu]\alpha^-$ is also inferred by ($\text{VAR}_-^{\simeq^D}$), since $[\mu]\alpha^-$ is a variable.

Case 3. ($\rightarrow^{\simeq^D}$)

Then we are proving that $P_1 \rightarrow M_1 \simeq^D P_2 \rightarrow M_2$ implies $[\mu](P_1 \rightarrow M_1) \simeq^D [\mu](P_2 \rightarrow M_2)$ (preserving the tree structure).

By inversion, we have $P_1 \simeq^D P_2$ and $M_1 \simeq^D M_2$, and thus, by the induction hypothesis, $[\mu]P_1 \simeq^D [\mu]P_2$ and $[\mu]M_1 \simeq^D [\mu]M_2$. Then $[\mu](P_1 \rightarrow M_1) \simeq^D [\mu](P_2 \rightarrow M_2)$, or in other words, $[\mu]P_1 \rightarrow [\mu]M_1 \simeq^D [\mu]P_2 \rightarrow [\mu]M_2$, is inferred by the same rule—($\rightarrow^{\simeq^D}$).

Case 4. ($\uparrow^{\simeq^D}$) This case is done by similar congruent arguments as the previous one.

Case 5. The positive cases are proved symmetrically.

□

Lemma 24. *The set of free variables is invariant under equivalence.*

– *If $N \simeq^D M$ then $\mathbf{fv}\, N = \mathbf{fv}\, M$ (as sets)*
+ *If $P \simeq^D Q$ then $\mathbf{fv}\, P = \mathbf{fv}\, Q$ (as sets)*

Proof. Mutual induction on $N \simeq^D M$ and $P \simeq^D Q$ The base cases (($\text{VAR}_-^{\simeq^D}$) and ($\text{VAR}_+^{\simeq^D}$)) are trivial. So are ($\uparrow^{\simeq^D}$), ($\downarrow^{\simeq^D}$), and ($\rightarrow^{\simeq^D}$), where the required property follows from the induction hypothesis.

Let us consider the case when the equivalence is formed by ($\forall^{\simeq^D}$), that is the equivalence has a shape $\forall\overrightarrow{\alpha^+}.\, N \simeq^D \forall\overrightarrow{\beta^+}.\, M$, and by inversion, there is a bijection $\mu : (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M) \leftrightarrow (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N)$ such that $N \simeq^D [\mu]M$, which by the induction hypothesis means $\mathbf{fv}\, N = \mathbf{fv}\, [\mu]M = [\mu]\mathbf{fv}\, M$.

Let us ensure by alpha-equivalence that $\overrightarrow{\alpha^+}$ is disjoint from $\mathbf{fv}\, M$. Then $(\mathbf{fv}\,\forall\overrightarrow{\beta^+}.\, M) \setminus \overrightarrow{\alpha^+} = \mathbf{fv}\,\forall\overrightarrow{\beta^+}.\, M$. Then we apply the following chain of equalities: $\mathbf{fv}\,\forall\overrightarrow{\alpha^+}.\, N = \mathbf{fv}\, N \setminus \overrightarrow{\alpha^+} = ([\mu]\mathbf{fv}\, M) \setminus \overrightarrow{\alpha^+} = [\mu](\mathbf{fv}\,\forall\overrightarrow{\beta^+}.\, M \cup (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M)) \setminus \overrightarrow{\alpha^+} = ([\mu]\mathbf{fv}\,\forall\overrightarrow{\beta^+}.\, M \cup [\mu](\overrightarrow{\beta^+} \cap \mathbf{fv}\, M)) \setminus \overrightarrow{\alpha^+} = ([\mu]\mathbf{fv}\,\forall\overrightarrow{\beta^+}.\, M) \setminus \overrightarrow{\alpha^+} = (\mathbf{fv}\,\forall\overrightarrow{\beta^+}.\, M) \setminus \overrightarrow{\alpha^+} = \mathbf{fv}\,\forall\overrightarrow{\beta^+}.\, M$.

Symmetrically, we prove the case when the equivalence is formed by ($\exists^{\simeq^D}$). □

Lemma 25 (Declarative equivalence is transitive).

+ *if $P_1 \simeq^D P_2$ and $P_2 \simeq^D P_3$ then $P_1 \simeq^D P_3$,*
– *if $N_1 \simeq^D N_2$ and $N_2 \simeq^D N_3$ then $N_1 \simeq^D N_3$.*

Proof. We prove it by $\text{size}(P_1 \simeq^D P_2) + \text{size}(P_2 \simeq^D P_3)$ and mutually, $\text{size}(N_1 \simeq^D N_2) + \text{size}(N_2 \simeq^D N_3)$, where by size, we mean the size of the nodes in the corresponding inference tree.

Case 1. First, let us consider the case when either $N_1 \simeq^D N_2$ or $N_2 \simeq^D N_3$ is inferred by ($\forall^{\simeq^D}$). Let us decompose N_1, N_2, and N_3 as follows: $N_1 = \forall\overrightarrow{\alpha^+}_1.\, M_1$, $N_2 = \forall\overrightarrow{\alpha^+}_2.\, M_2$, and $N_3 = \forall\overrightarrow{\alpha^+}_3.\, M_3$.

Then by inversion of $\forall\overrightarrow{\alpha^+}_1.\, M_1 \simeq^D \forall\overrightarrow{\alpha^+}_2.\, M_2$ (or if $\overrightarrow{\alpha^+}_1$ and $\overrightarrow{\alpha^+}_2$ are both empty, by assumption) :

(1) $\overrightarrow{\alpha^+}_1 \cap \mathbf{fv}\, M_2 = \emptyset$ and

(2) there exists a bijection on variables $\mu_1 : (\overrightarrow{\alpha^+}_2 \cap \mathbf{fv}\, M_2) \leftrightarrow (\overrightarrow{\alpha^+}_1 \cap \mathbf{fv}\, M_1)$ such that $M_1 \simeq^D [\mu_1]M_2$.

Analogously, $\forall\overrightarrow{\alpha^+}_1.\ M_1 \simeq^D \forall\overrightarrow{\alpha^+}_2.\ M_2$ implies:

(1) $\overrightarrow{\alpha^+}_2 \cap \mathbf{fv}\, M_3 = \emptyset$ and

(2) $M_2 \simeq^D [\mu_2]M_3$ for some bijection $\mu_2 : (\overrightarrow{\alpha^+}_3 \cap \mathbf{fv}\, M_3) \leftrightarrow (\overrightarrow{\alpha^+}_2 \cap \mathbf{fv}\, M_2)$.

Notice that either $M_1 \simeq^D [\mu_1]M_2$ is inferred by a proper sub-tree of $\forall\overrightarrow{\alpha^+}_1.\ M_1 \simeq^D \forall\overrightarrow{\alpha^+}_2.\ M_2$ or $M_2 \simeq^D [\mu_2]M_3$ is inferred by a proper sub-tree of $\forall\overrightarrow{\alpha^+}_2.\ M_2 \simeq^D \forall\overrightarrow{\alpha^+}_3.\ M_3$.

Then by lemma 23, $[\mu_1]M_2 \simeq^D [\mu_1 \circ \mu_2]M_3$ and moreover, $\mathrm{size}([\mu_1]M_2 \simeq^D [\mu_1 \circ \mu_2]M_3) = \mathrm{size}(M_2 \simeq^D [\mu_2]M_3)$.

Since at least one of the trees inferring $M_1 \simeq^D [\mu_1]M_2$ and $M_2 \simeq^D [\mu_2]M_3$ is a proper sub-tree of the corresponding original tree, $\mathrm{size}(M_1 \simeq^D [\mu_1]M_2) + \mathrm{size}(M_2 \simeq^D [\mu_2]M_3) < \mathrm{size}(\forall\overrightarrow{\alpha^+}_1.\ M_1 \simeq^D \forall\overrightarrow{\alpha^+}_2.\ M_2) + \mathrm{size}(\forall\overrightarrow{\alpha^+}_2.\ M_2 \simeq^D \forall\overrightarrow{\alpha^+}_3.\ M_3)$, i.e., the induction hypothesis is applicable.

By the induction hypothesis, $M_1 \simeq^D [\mu_1 \circ \mu_2]M_3$. Where $\mu_1 \circ \mu_2$ is a bijection on variables $\mu_1 \circ \mu_2 : (\overrightarrow{\alpha^+}_3 \cap \mathbf{fv}\, M_3) \leftrightarrow (\overrightarrow{\alpha^+}_1 \cap \mathbf{fv}\, M_1)$. Then $\forall\overrightarrow{\alpha^+}_1.\ M_1 \simeq^D \forall\overrightarrow{\alpha^+}_3.\ M_3$ by $(\forall^{\simeq^D})$.

Once this case has been considered, we can assume that neither $N_1 \simeq^D N_2$ nor $N_2 \simeq^D N_3$ is inferred by $(\forall^{\simeq^D})$.

Case 2. $N_1 \simeq^D N_2$ is inferred by $(\mathrm{VAR}_-^{\simeq^D})$

Then $N_1 = N_2 = \alpha^-$, and thus, $N_1 \simeq^D N_3$ holds since $N_2 \simeq^D N_3$.

Case 3. $N_1 \simeq^D N_2$ is inferred by $(\rightarrow^{\simeq^D})$

Then $N_1 = P_1 \rightarrow M_1$ and $N_2 = P_2 \rightarrow M_2$, and by inversion, $P_1 \simeq^D P_2$ and $M_1 \simeq^D M_2$.

Moreover, since N_3 does not start with $\forall$, $N_2 \simeq^D N_3$ is also inferred by $(\rightarrow^{\simeq^D})$, which means that $N_3 = P_3 \rightarrow M_3$, $P_2 \simeq^D P_3$, and $M_2 \simeq^D M_3$.

Then by the induction hypothesis, $P_1 \simeq^D P_3$ and $M_1 \simeq^D M_3$, and thus, $P_1 \rightarrow M_1 \simeq^D P_3 \rightarrow M_3$ by $(\rightarrow^{\simeq^D})$.

Case 4. $N_1 \simeq^D N_2$ is inferred by $(\rightarrow^{\simeq^D})$

For this case, the reasoning is the same as for the previous one.

Case 5. The positive cases are proved symmetrically.

□

Lemma 26 (Type well-formedness is invariant under equivalence). *Mutual subtyping implies declarative equivalence.*

+ *if* $P \simeq^D Q$ *then* $T \vdash P \iff T \vdash Q$,
− *if* $N \simeq^D M$ *then* $T \vdash N \iff T \vdash M$

Proof. We prove it by induction on $P \simeq^D Q$ and mutually, on $N \simeq^D M$. Let us consider the last rule used in the derivation.

Case 1. $(\mathrm{VAR}_-^{\simeq^D})$, that is $N \simeq^D M$ has shape $\alpha^- \simeq^D \alpha^-$.

Than $T \vdash P \iff T \vdash Q$ is rewritten as $T \vdash \alpha^- \iff T \vdash \alpha^-$, which holds trivially.

Case 2. $(\uparrow^{\simeq^D})$, that is $N \simeq^D M$ has shape $\uparrow P \simeq^D \uparrow Q$.

By inversion, $P \simeq^D Q$, and by the induction hypothesis, $T \vdash P \iff T \vdash Q$. Also notice that $T \vdash \uparrow P \iff T \vdash P$ and $T \vdash \uparrow Q \iff T \vdash Q$ by inversion and $(\uparrow^{\mathrm{WF}})$. This way, $T \vdash \uparrow P \iff T \vdash P \iff T \vdash Q \iff T \vdash \uparrow Q$.

Case 3. $(\rightarrow^{\simeq^D})$, that is $N \simeq^D M$ has shape $P \rightarrow N' \simeq^D Q \rightarrow M'$.

Then by inversion, $P \simeq^D Q$ and $N' \simeq^D M'$, and by the induction hypothesis, $T \vdash P \iff T \vdash Q$ and $T \vdash N' \iff T \vdash M'$.

$$\begin{aligned} T \vdash P \to N' &\iff T \vdash P \text{ and } T \vdash N' && \text{by inversion and } (\to^{\mathrm{WF}}) \\ &\iff T \vdash Q \text{ and } T \vdash M' && \text{as noted above} \\ &\iff T \vdash Q \to M' && \text{by } (\to^{\mathrm{WF}}) \text{ and inversion} \end{aligned}$$

Case 4. $(\forall^{\simeq^D})$, that is $N \simeq^D M$ has shape $\forall \overrightarrow{\alpha^+}.\ N' \simeq^D \forall \overrightarrow{\beta^+}.\ M'$.

By inversion, $\forall \overrightarrow{\alpha^+}.\ N' \simeq^D \forall \overrightarrow{\beta^+}.\ M'$ means that $\overrightarrow{\alpha^+} \cap \mathbf{fv}\, M = \emptyset$ and that there exists a bijection on variables $\mu : (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M') \leftrightarrow (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N')$ such that $N' \simeq^D [\mu]M'$.

By inversion and $(\forall^{\mathrm{WF}})$, $T \vdash \forall \overrightarrow{\alpha^+}.\ N'$ is equivalent to $T, \overrightarrow{\alpha^+} \vdash N'$, and by lemma 2, it is equivalent to $T, (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N') \vdash N'$, which, by the induction hypothesis, is equivalent to $T, (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N') \vdash [\mu]M'$.

Analogously, $T \vdash \forall \overrightarrow{\beta^+}.\ M'$ is equivalent to $T, (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M') \vdash M'$. By lemma 3, it implies $T, (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, M') \vdash [\mu]M'$. And vice versa, $T, (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, M') \vdash [\mu]M'$ implies $T, (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M') \vdash [\mu^{-1}][\mu]M'$.

This way, both $T \vdash \forall \overrightarrow{\alpha^+}.\ N'$ and $T \vdash \forall \overrightarrow{\beta^+}.\ M'$ are equivalent to $T, (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N') \vdash [\mu]M'$.

Case 5. For the cases of the positive types, the proofs are symmetric.

□

Lemma 27 (Soundness of equivalence). *Declarative equivalence implies mutual subtyping.*

+ *if* $T \vdash P$, $T \vdash Q$, *and* $P \simeq^D Q$ *then* $T \vdash P \simeq^{\leqslant} Q$,
− *if* $T \vdash N$, $T \vdash M$, *and* $N \simeq^D M$ *then* $T \vdash N \simeq^{\leqslant} M$.

Proof. We prove it by mutual induction on $P \simeq^D Q$ and $N \simeq^D M$.

Case 1. $\alpha^- \simeq^D \alpha^-$

Then $T \vdash \alpha^- \leqslant \alpha^-$ by $(\mathrm{Var}^{\leqslant}_-)$, which immediately implies $T \vdash \alpha^- \simeq^{\leqslant} \alpha^-$ by $(\simeq^{\leqslant}_-)$.

Case 2. $\uparrow P \simeq^D \uparrow Q$

Then by inversion of $(\uparrow^{\leqslant})$, $P \simeq^D Q$, and from the induction hypothesis, $T \vdash P \simeq^{\leqslant} Q$, and (by symmetry) $T \vdash Q \simeq^{\leqslant} P$.

When $(\uparrow^{\leqslant})$ is applied to $T \vdash P \simeq^{\leqslant} Q$, it gives us $T \vdash \uparrow P \leqslant \uparrow Q$; when it is applied to $T \vdash Q \simeq^{\leqslant} P$, we obtain $T \vdash \uparrow Q \leqslant \uparrow P$. Together, it implies $T \vdash \uparrow P \simeq^{\leqslant} \uparrow Q$.

Case 3. $P \to N \simeq^D Q \to M$

Then by inversion of $(\to^{\leqslant})$, $P \simeq^D Q$ and $N \simeq^D M$. By the induction hypothesis, $T \vdash P \simeq^{\leqslant} Q$ and $T \vdash N \simeq^{\leqslant} M$, which means by inversion: (i) $T \vdash P \geqslant Q$, (ii) $T \vdash Q \geqslant P$, (iii) $T \vdash N \leqslant M$, (iv) $T \vdash M \leqslant N$. Applying $(\to^{\leqslant})$ to (i) and (iii), we obtain $T \vdash P \to N \leqslant Q \to M$; applying it to (ii) and (iv), we have $T \vdash Q \to M \leqslant P \to N$. Together, it implies $T \vdash P \to N \simeq^{\leqslant} Q \to M$.

Case 4. $\forall \overrightarrow{\alpha^+}.\ N \simeq^D \forall \overrightarrow{\beta^+}.\ M$

Then by inversion, there exists bijection $\mu : (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M) \leftrightarrow (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N)$, such that $N \simeq^D [\mu]M$. By the induction hypothesis, $T, \overrightarrow{\alpha^+} \vdash N \simeq^{\leqslant} [\mu]M$. From corollary 9 and the fact that μ is bijective, we also have $T, \overrightarrow{\beta^+} \vdash [\mu^{-1}]N \simeq^{\leqslant} M$.

Let us construct a substitution $\overrightarrow{\alpha^+} \vdash \overrightarrow{P}/\overrightarrow{\beta^+} : \overrightarrow{\beta^+}$ by extending μ with arbitrary positive types on $\overrightarrow{\beta^+} \setminus \mathbf{fv}\, M$.

Notice that $[\mu]M = [\overrightarrow{P}/\overrightarrow{\beta^+}]M$, and therefore, $T, \overrightarrow{\alpha^+} \vdash N \simeq^{\leqslant} [\mu]M$ implies $T, \overrightarrow{\alpha^+} \vdash [\overrightarrow{P}/\overrightarrow{\beta^+}]M \leqslant N$. Then by $(\forall^{\leqslant})$, $T \vdash \forall \overrightarrow{\beta^+}.\ M \leqslant \forall \overrightarrow{\alpha^+}.\ N$.

Analogously, we construct the substitution from μ^{-1}, and use it to instantiate $\overrightarrow{\alpha^+}$ in the application of $(\forall^{\leqslant})$ to infer $T \vdash \forall \overrightarrow{\alpha^+}.\ N \leqslant \forall \overrightarrow{\beta^+}.\ M$.

This way, $T \vdash \forall \overrightarrow{\beta^+}.\ M \leqslant \forall \overrightarrow{\alpha^+}.\ N$ and $T \vdash \forall \overrightarrow{\alpha^+}.\ N \leqslant \forall \overrightarrow{\beta^+}.\ M$ gives us $T \vdash \forall \overrightarrow{\beta^+}.\ M \simeq^{\leqslant} \forall \overrightarrow{\alpha^+}.\ N$.

Case 5. For the cases of the positive types, the proofs are symmetric.

□

Lemma 28 (Subtyping induced by disjoint substitutions). *Suppose that $T \vdash \sigma_1 : T_1$ and $T \vdash \sigma_2 : T_1$, where $T_i \subseteq T$ and $T_1 \cap T_2 = \emptyset$. Then*

- *assuming $T \vdash N$, $T \vdash [\sigma_1]N \leqslant [\sigma_2]N$ implies $T \vdash \sigma_i \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, N$*

+ *assuming $T \vdash P$, $T \vdash [\sigma_1]P \geqslant [\sigma_2]P$ implies $T \vdash \sigma_i \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, P$*

Proof. Proof by induciton on $T \vdash N$ (and mutually on $T \vdash P$).

Case 1. $N = \alpha^-$

Then $T \vdash [\sigma_1]N \leqslant [\sigma_2]N$ is rewritten as $T \vdash [\sigma_1]\alpha^- \leqslant [\sigma_2]\alpha^-$. Let us consider the following cases:

a. $\alpha^- \notin T_1$ and $\alpha^- \notin T_2$

Then $T \vdash \sigma_i \simeq^{\leqslant} \mathrm{id} : \alpha^-$ holds immediately, since $[\sigma_i]\alpha^- = [\mathrm{id}]\alpha^- = \alpha^-$ and $T \vdash \alpha^- \simeq^{\leqslant} \alpha^-$.

b. $\alpha^- \in T_1$ and $\alpha^- \in T_2$

This case is not possible by assumption: $T_1 \cap T_2 = \emptyset$.

c. $\alpha^- \in T_1$ and $\alpha^- \notin T_2$

Then we have $T \vdash [\sigma_1]\alpha^- \leqslant \alpha^-$, which by corollary 8 means $T \vdash [\sigma_1]\alpha^- \simeq^{\leqslant} \alpha^-$, and hence, $T \vdash \sigma_1 \simeq^{\leqslant} \mathrm{id} : \alpha^-$.

$T \vdash \sigma_2 \simeq^{\leqslant} \mathrm{id} : \alpha^-$ holds since $[\sigma_2]\alpha^- = \alpha^-$, similarly to case 1.*a*.

d. $\alpha^- \notin T_1$ and $\alpha^- \in T_2$

Then we have $T \vdash \alpha^- \leqslant [\sigma_2]\alpha^-$, which by corollary 8 means $T \vdash \alpha^- \simeq^{\leqslant} [\sigma_2]\alpha^-$, and hence, $T \vdash \sigma_2 \simeq^{\leqslant} \mathrm{id} : \alpha^-$.

$T \vdash \sigma_1 \simeq^{\leqslant} \mathrm{id} : \alpha^-$ holds since $[\sigma_1]\alpha^- = \alpha^-$, similarly to case 1.*a*.

Case 2. $N = \forall\overrightarrow{\alpha^+}.\, M$

Then by inversion, $T, \overrightarrow{\alpha^+} \vdash M$. $T \vdash [\sigma_1]N \leqslant [\sigma_2]N$ is rewritten as $T \vdash [\sigma_1]\forall\overrightarrow{\alpha^+}.\, M \leqslant [\sigma_2]\forall\overrightarrow{\alpha^+}.\, M$. By the congruence of substitution and by the inversion of $(\forall^{\leqslant})$, $T, \overrightarrow{\alpha^+} \vdash [\overrightarrow{Q}/\overrightarrow{\alpha^+}][\sigma_1]M \leqslant [\sigma_2]M$, where $T, \overrightarrow{\alpha^+} \vdash Q_i$. Let us denote the (Kleisli) composition of σ_1 and $\overrightarrow{Q}/\overrightarrow{\alpha^+}$ as σ_1', noting that $T, \overrightarrow{\alpha^+} \vdash \sigma_1' : T_1, \overrightarrow{\alpha^+}$, and $(T_1, \overrightarrow{\alpha^+}) \cap T_2 = \emptyset$.

Let us apply the induction hypothesis to M and the substitutions σ_1' and σ_2 with $T, \overrightarrow{\alpha^+} \vdash [\sigma_1']M \leqslant [\sigma_2]M$ to obtain:

$$T, \overrightarrow{\alpha^+} \vdash \sigma_1' \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, M \tag{1}$$

$$T, \overrightarrow{\alpha^+} \vdash \sigma_2 \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, M \tag{2}$$

Then $T \vdash \sigma_2 \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, \forall\overrightarrow{\alpha^+}.\, M$ holds by strengthening of 2: for any $\beta^{\pm} \in \mathbf{fv}\, \forall\overrightarrow{\alpha^+}.\, M = \mathbf{fv}\, M \setminus \overrightarrow{\alpha^+}$, $T, \overrightarrow{\alpha^+} \vdash [\sigma_2]\beta^{\pm} \simeq^{\leqslant} \beta^{\pm}$ is strengthened to $T \vdash [\sigma_2]\beta^{\pm} \simeq^{\leqslant} \beta^{\pm}$, because $\mathbf{fv}\, [\sigma_2]\beta^{\pm} = \mathbf{fv}\, \beta^{\pm} = \{\beta^{\pm}\} \subseteq T$.

To show that $T \vdash \sigma_1 \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, \forall\overrightarrow{\alpha^+}.\, M$, let us take an arbitrary $\beta^{\pm} \in \mathbf{fv}\, \forall\overrightarrow{\alpha^+}.\, M = \mathbf{fv}\, M \setminus \overrightarrow{\alpha^+}$.

$$\begin{aligned}
\beta^{\pm} &= [\mathrm{id}]\beta^{\pm} && \text{by definition of } \mathbf{id}\\
&\simeq^{\leqslant} [\sigma_1']\beta^{\pm} && \text{by 1}\\
&= [\overrightarrow{Q}/\overrightarrow{\alpha^+}][\sigma_1]\beta^{\pm} && \text{by definition of } \sigma_1'\\
&= [\sigma_1]\beta^{\pm} && \text{because } \overrightarrow{\alpha^+} \cap \mathbf{fv}\, [\sigma_1]\beta^{\pm} \subseteq \overrightarrow{\alpha^+} \cap T = \emptyset
\end{aligned}$$

This way, $T \vdash [\sigma_1]\beta^{\pm} \simeq^{\leqslant} \beta^{\pm}$ for any $\beta^{\pm} \in \mathbf{fv}\, \forall\overrightarrow{\alpha^+}.\, M$ and thus, $T \vdash \sigma_1 \simeq^{\leqslant} \mathrm{id} : \mathbf{fv}\, \forall\overrightarrow{\alpha^+}.\, M$.

Case 3. $N = P \to M$

Then by inversion, $T \vdash P$ and $T \vdash M$. $T \vdash [\sigma_1]N \leqslant [\sigma_2]N$ is rewritten as $T \vdash [\sigma_1](P \to M) \leqslant$

$[\sigma_2](P \to M)$, then by congruence of substitution, $T \vdash [\sigma_1]P \to [\sigma_1]M \leqslant [\sigma_2]P \to [\sigma_2]M$, then by inversion $T \vdash [\sigma_1]P \geqslant [\sigma_2]P$ and $T \vdash [\sigma_1]M \leqslant [\sigma_2]M$.
Applying the induction hypothesis to $T \vdash [\sigma_1]P \geqslant [\sigma_2]P$ and to $T \vdash [\sigma_1]M \leqslant [\sigma_2]M$, we obtain (respectively):

$$T \vdash \sigma_i \simeq^{\leqslant} \text{id} : \mathbf{fv}\, P \tag{3}$$

$$T \vdash \sigma_i \simeq^{\leqslant} \text{id} : \mathbf{fv}\, M \tag{4}$$

Noting that $\mathbf{fv}\,(P \to M) = \mathbf{fv}\, P \cup \mathbf{fv}\, M$, we combine eqs. (3) and (4) to conclude: $T \vdash \sigma_i \simeq^{\leqslant} \text{id} : \mathbf{fv}\,(P \to M)$.

Case 4. $N = \uparrow P$
Then by inversion, $T \vdash P$. $T \vdash [\sigma_1]N \leqslant [\sigma_2]N$ is rewritten as $T \vdash [\sigma_1]{\uparrow}P \leqslant [\sigma_2]{\uparrow}P$, then by congruence of substitution and by inversion, $T \vdash [\sigma_1]P \geqslant [\sigma_2]P$
Applying the induction hypothesis to $T \vdash [\sigma_1]P \geqslant [\sigma_2]P$, we obtain $T \vdash \sigma_i \simeq^{\leqslant} \text{id} : \mathbf{fv}\, P$. Since $\mathbf{fv}\,{\uparrow}P = \mathbf{fv}\, P$, we can conclude: $T \vdash \sigma_i \simeq^{\leqslant} \text{id} : \mathbf{fv}\,{\uparrow}P$.

Case 5. The positive cases are proved symmetrically.

□

Corollary 11 (Substitution cannot induce proper subtypes or supertypes). *Assuming all mentioned types are well-formed in T and σ is a substitution $T \vdash \sigma : T$,*

$$\begin{aligned}
T \vdash [\sigma]N \leqslant N &\implies T \vdash [\sigma]N \simeq^{\leqslant} N \text{ and } T \vdash \sigma \simeq^{\leqslant} \text{id} : \mathbf{fv}\, N\\
T \vdash N \leqslant [\sigma]N &\implies T \vdash N \simeq^{\leqslant} [\sigma]N \text{ and } T \vdash \sigma \simeq^{\leqslant} \text{id} : \mathbf{fv}\, N\\
T \vdash [\sigma]P \geqslant P &\implies T \vdash [\sigma]P \simeq^{\leqslant} P \text{ and } T \vdash \sigma \simeq^{\leqslant} \text{id} : \mathbf{fv}\, P\\
T \vdash P \geqslant [\sigma]P &\implies T \vdash P \simeq^{\leqslant} [\sigma]P \text{ and } T \vdash \sigma \simeq^{\leqslant} \text{id} : \mathbf{fv}\, P
\end{aligned}$$

Lemma 29 (Mutual substitution and subtyping). *Assuming that the mentioned types (P, Q, N, and M) are well-formed in T and that the substitutions (σ_1 and σ_2) have signature $T \vdash \sigma_i : T$,*

+ *if $T \vdash [\sigma_1]P \geqslant Q$ and $T \vdash [\sigma_2]Q \geqslant P$*
  then there exists a bijection $\mu : \mathbf{fv}\, P \leftrightarrow \mathbf{fv}\, Q$ such that $T \vdash \sigma_1 \simeq^{\leqslant} \mu : \mathbf{fv}\, P$ and $T \vdash \sigma_2 \simeq^{\leqslant} \mu^{-1} : \mathbf{fv}\, Q$;
− *if $T \vdash [\sigma_1]N \leqslant M$ and $T \vdash [\sigma_2]N \leqslant M$*
  then there exists a bijection $\mu : \mathbf{fv}\, N \leftrightarrow \mathbf{fv}\, M$ such that $T \vdash \sigma_1 \simeq^{\leqslant} \mu : \mathbf{fv}\, N$ and $T \vdash \sigma_2 \simeq^{\leqslant} \mu^{-1} : \mathbf{fv}\, M$.

Proof.

+ Applying σ_2 to both sides of $T \vdash [\sigma_1]P \geqslant Q$ (by lemma 21), we have: $T \vdash [\sigma_2 \circ \sigma_1]P \geqslant [\sigma_2]Q$. Composing it with $T \vdash [\sigma_2]Q \geqslant P$ by transitivity (lemma 22), we have $T \vdash [\sigma_2 \circ \sigma_1]P \geqslant P$. Then by corollary 11, $T \vdash \sigma_2 \circ \sigma_1 \simeq^{\leqslant} \text{id} : \mathbf{fv}\, P$. By a symmetric argument, we also have: $T \vdash \sigma_1 \circ \sigma_2 \simeq^{\leqslant} \text{id} : \mathbf{fv}\, Q$.
  Now, we prove that $T \vdash \sigma_2 \circ \sigma_1 \simeq^{\leqslant} \text{id} : \mathbf{fv}\, P$ and $T \vdash \sigma_1 \circ \sigma_2 \simeq^{\leqslant} \text{id} : \mathbf{fv}\, Q$ implies that σ_1 and σ_1 are (equivalent to) mutually inverse bijections.
  To do so, it suffices to prove that
  (i) for any $\alpha^{\pm} \in \mathbf{fv}\, P$ there exists $\beta^{\pm} \in \mathbf{fv}\, Q$ such that $T \vdash [\sigma_1]\alpha^{\pm} \simeq^{\leqslant} \beta^{\pm}$ and $T \vdash [\sigma_2]\beta^{\pm} \simeq^{\leqslant} \alpha^{\pm}$; and
  (ii) for any $\beta^{\pm} \in \mathbf{fv}\, Q$ there exists $\alpha^{\pm} \in \mathbf{fv}\, P$ such that $T \vdash [\sigma_2]\beta^{\pm} \simeq^{\leqslant} \alpha^{\pm}$ and $T \vdash [\sigma_1]\alpha^{\pm} \simeq^{\leqslant} \beta^{\pm}$.

Then these correspondences between **fv** P and **fv** Q are mutually inverse functions, since for any $\beta^\pm$ there can be at most one $\alpha^\pm$ such that $T \vdash [\sigma_2]\beta^\pm \simeq^\leqslant \alpha^\pm$ (and vice versa).

(i) Let us take $\alpha^\pm \in \mathbf{fv}\, P$.

(a) if $\alpha^\pm$ is positive ($\alpha^\pm = \alpha^+$), from $T \vdash [\sigma_2][\sigma_1]\alpha^+ \simeq^\leqslant \alpha^+$, by corollary 8, we have $[\sigma_2][\sigma_1]\alpha^+ = \exists\overrightarrow{\beta^-}.\, \alpha^+$.
What shape can $[\sigma_1]\alpha^+$ have? It cannot be $\exists\overrightarrow{\alpha^-}.\, {\downarrow}N$ (for potentially empty $\overrightarrow{\alpha^-}$), because the outer constructor $\downarrow$ would remain after the substitution σ_2, whereas $\exists\overrightarrow{\beta^-}.\, \alpha^+$ does not have $\downarrow$. The only case left is $[\sigma_1]\alpha^+ = \exists\overrightarrow{\alpha^-}.\, \gamma^+$.
Notice that $T \vdash \exists\overrightarrow{\alpha^-}.\, \gamma^+ \simeq^\leqslant \gamma^+$, meaning that $T \vdash [\sigma_1]\alpha^+ \simeq^\leqslant \gamma^+$. Also notice that $[\sigma_2]\exists\overrightarrow{\alpha^-}.\, \gamma^+ = \exists\overrightarrow{\beta^-}.\, \alpha^+$ implies $T \vdash [\sigma_2]\gamma^+ \simeq^\leqslant \alpha^+$.

(b) if $\alpha^\pm$ is negative ($\alpha^\pm = \alpha^-$) from $T \vdash [\sigma_2][\sigma_1]\alpha^- \simeq^\leqslant \alpha^-$, by corollary 8, we have $[\sigma_2][\sigma_1]\alpha^- = \forall\overrightarrow{\beta^+}.\, \alpha^-$.
What shape can $[\sigma_1]\alpha^-$ have? It cannot be $\forall\overrightarrow{\alpha^+}.\, {\uparrow}P$ nor $\forall\overrightarrow{\alpha^+}.\, P \to M$ (for potentially empty $\overrightarrow{\alpha^+}$), because the outer constructor ($\to$ or $\uparrow$), remaining after the substitution σ_2, is however absent in the resulting $\forall\overrightarrow{\beta^+}.\, \alpha^-$. Hence, the only case left is $[\sigma_1]\alpha^- = \forall\overrightarrow{\alpha^+}.\, \gamma^-$ Notice that $T \vdash \gamma^- \simeq^\leqslant \forall\overrightarrow{\alpha^+}.\, \gamma^-$, meaning that $T \vdash [\sigma_1]\alpha^- \simeq^\leqslant \gamma^-$.
Also notice that $[\sigma_2]\forall\overrightarrow{\alpha^+}.\, \gamma^- = \forall\overrightarrow{\beta^+}.\, \alpha^-$ implies $T \vdash [\sigma_2]\gamma^- \simeq^\leqslant \alpha^-$.

(ii) The proof is symmetric: We swap P and Q, σ_1 and σ_2, and exploit $T \vdash [\sigma_1][\sigma_2]\alpha^\pm \simeq^\leqslant \alpha^\pm$ instead of $T \vdash [\sigma_2][\sigma_1]\alpha^\pm \simeq^\leqslant \alpha^\pm$.

– The proof is symmetric to the positive case.

□

Lemma 30 (Equivalent substitution act equivalently). *Suppose that $T' \vdash \sigma_1 : T$ and $T' \vdash \sigma_2 : T$ are substitutions equivalent on their domain, that is $T' \vdash \sigma_1 \simeq^\leqslant \sigma_2 : T$. Then*

+ *for any $T \vdash P$, $T' \vdash [\sigma_1]P \simeq^\leqslant [\sigma_2]P$;*
– *for any $T \vdash N$, $T' \vdash [\sigma_1]N \simeq^\leqslant [\sigma_2]N$.*

Proof. We prove it by induction on P (and mutually on N).

Case 1. $N = \alpha^-$
Then since by inversion, $\alpha^- \in T$, $T' \vdash [\sigma_1]\alpha^- \simeq^\leqslant [\sigma_2]\alpha^-$ holds by definition of $T' \vdash \sigma_1 \simeq^\leqslant \sigma_2 : T$.

Case 2. $N = {\uparrow}P$
Then by inversion, $T \vdash P$. By the induction hypothesis, $T' \vdash [\sigma_1]P \simeq^\leqslant [\sigma_2]P$, Then by $(\uparrow^\leqslant)$, $T' \vdash {\uparrow}[\sigma_1]P \leqslant {\uparrow}[\sigma_2]P$, and symmetrically, $T' \vdash {\uparrow}[\sigma_2]P \leqslant {\uparrow}[\sigma_1]P$, together meaning that $T' \vdash {\uparrow}[\sigma_1]P \simeq^\leqslant {\uparrow}[\sigma_2]P$, or equivalently, $T' \vdash [\sigma_1]{\uparrow}P \simeq^\leqslant [\sigma_2]{\uparrow}P$.

Case 3. $N = P \to M$
Then by inversion, $T \vdash P$ and $T \vdash M$. By the induction hypothesis, $T' \vdash [\sigma_1]P \simeq^\leqslant [\sigma_2]P$ and $T' \vdash [\sigma_1]M \simeq^\leqslant [\sigma_2]M$, that is $T' \vdash [\sigma_1]P \geqslant [\sigma_2]P$, $T' \vdash [\sigma_2]P \geqslant [\sigma_1]P$, $T' \vdash [\sigma_1]M \leqslant [\sigma_2]M$, and $T' \vdash [\sigma_2]M \leqslant [\sigma_1]M$. Then by $(\to^\leqslant)$, $T' \vdash [\sigma_1]P \to [\sigma_1]M \leqslant [\sigma_2]P \to [\sigma_2]M$, and again by $(\to^\leqslant)$, $T' \vdash [\sigma_2]P \to [\sigma_2]M \leqslant [\sigma_1]P \to [\sigma_1]M$. This way, $T' \vdash [\sigma_1]P \to [\sigma_1]M \simeq^\leqslant [\sigma_2]P \to [\sigma_2]M$, or equivalently, $T' \vdash [\sigma_1](P \to M) \simeq^\leqslant [\sigma_2](P \to M)$.

Case 4. $N = \forall\overrightarrow{\alpha^+}.\, M$ We can assume that $\overrightarrow{\alpha^+}$ is disjoint from T and T'. By inversion, $T \vdash \forall\overrightarrow{\alpha^+}.\, M$ implies $T, \overrightarrow{\alpha^+} \vdash M$. Notice that $T' \vdash \sigma_i : T$ and $T' \vdash \sigma_1 \simeq^\leqslant \sigma_2 : T$ can be extended to $T', \overrightarrow{\alpha^+} \vdash \sigma_i : T, \overrightarrow{\alpha^+}$ and $T', \overrightarrow{\alpha^+} \vdash \sigma_1 \simeq^\leqslant \sigma_2 : T, \overrightarrow{\alpha^+}$ by lemma 12. Then by the induction hypothesis, $T', \overrightarrow{\alpha^+} \vdash [\sigma_1]M \simeq^\leqslant [\sigma_2]M$, meaning by inversion that $T', \overrightarrow{\alpha^+} \vdash [\sigma_1]M \leqslant [\sigma_2]M$ and $T', \overrightarrow{\alpha^+} \vdash [\sigma_2]M \leqslant [\sigma_1]M$.

To infer $T' \vdash \forall\overrightarrow{\alpha^+}.\, [\sigma_1]M \leqslant \forall\overrightarrow{\alpha^+}.\, [\sigma_2]M$, we apply ($\forall^{\leqslant}$) with the substitution $T', \overrightarrow{\alpha^+} \vdash \mathrm{id} : \overrightarrow{\alpha^+}$, noting that $T', \overrightarrow{\alpha^+} \vdash [\mathrm{id}][\sigma_1]M \leqslant [\sigma_2]M$ holds since $T', \overrightarrow{\alpha^+} \vdash [\sigma_1]M \leqslant [\sigma_2]M$, as noted above.
Symmetrically, we infer $T' \vdash \forall\overrightarrow{\alpha^+}.\, [\sigma_2]M \leqslant \forall\overrightarrow{\alpha^+}.\, [\sigma_1]M$, which together with $T' \vdash \forall\overrightarrow{\alpha^+}.\, [\sigma_1]M \leqslant \forall\overrightarrow{\alpha^+}.\, [\sigma_2]M$ means $T' \vdash \forall\overrightarrow{\alpha^+}.\, [\sigma_1]M \simeq^{\leqslant} \forall\overrightarrow{\alpha^+}.\, [\sigma_2]M$, or equivalently, $T' \vdash [\sigma_1]\forall\overrightarrow{\alpha^+}.\, M \simeq^{\leqslant} [\sigma_2]\forall\overrightarrow{\alpha^+}.\, M$.

Case 5. The positive cases are proved symmetrically.

□

Lemma 31 (Equivalence of polymorphic types)**.**

− *For* $T \vdash \forall\overrightarrow{\alpha^+}.\, N$ *and* $T \vdash \forall\overrightarrow{\beta^+}.\, M$*,*
if $T \vdash \forall\overrightarrow{\alpha^+}.\, N \simeq^{\leqslant} \forall\overrightarrow{\beta^+}.\, M$ *then there exists a bijection* $\mu : \overrightarrow{\beta^+} \cap \mathbf{fv}\, M \leftrightarrow \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N$ *such that* $T, \overrightarrow{\alpha^+} \vdash N \simeq^{\leqslant} [\mu]M$*,*

+ *For* $T \vdash \exists\overrightarrow{\alpha^-}.\, P$ *and* $T \vdash \exists\overrightarrow{\beta^-}.\, Q$*,*
if $T \vdash \exists\overrightarrow{\alpha^-}.\, P \simeq^{\leqslant} \exists\overrightarrow{\beta^-}.\, Q$ *then there exists a bijection* $\mu : \overrightarrow{\beta^-} \cap \mathbf{fv}\, Q \leftrightarrow \overrightarrow{\alpha^-} \cap \mathbf{fv}\, P$ *such that* $T, \overrightarrow{\beta^-} \vdash P \simeq^{\leqslant} [\mu]Q$*.*

Proof.

− First, by α-conversion, we ensure $\overrightarrow{\alpha^+} \cap \mathbf{fv}\, M = \emptyset$ and $\overrightarrow{\beta^+} \cap \mathbf{fv}\, N = \emptyset$. By inversion, $T \vdash \forall\overrightarrow{\alpha^+}.\, N \simeq^{\leqslant} \forall\overrightarrow{\beta^+}.\, M$ implies
(1) $T, \overrightarrow{\beta^+} \vdash [\sigma_1]N \leqslant M$ for $T, \overrightarrow{\beta^+} \vdash \sigma_1 : \overrightarrow{\alpha^+}$ and
(2) $T, \overrightarrow{\alpha^+} \vdash [\sigma_2]M \leqslant N$ for $T, \overrightarrow{\alpha^+} \vdash \sigma_2 : \overrightarrow{\beta^+}$.
To apply lemma 29, we weaken and rearrange the contexts, and extend the substitutions to act as identity outside of their initial domain:
(1) $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\sigma_1]N \leqslant M$ for $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \sigma_1 : T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+}$ and
(2) $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\sigma_2]M \leqslant N$ for $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \sigma_2 : T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+}$.
Then from lemma 29, there exists a bijection $\mu : \mathbf{fv}\, M \leftrightarrow \mathbf{fv}\, N$ such that $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \sigma_2 \simeq^{\leqslant} \mu : \mathbf{fv}\, M$ and $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \sigma_1 \simeq^{\leqslant} \mu^{-1} : \mathbf{fv}\, N$.
Let us show that $\mu|_{\overrightarrow{\beta^+}}$ is the appropriate candidate.
First, we show that if we restrict the domain of μ to $\overrightarrow{\beta^+}$, its range will be contained in $\overrightarrow{\alpha^+}$. Let us take $\gamma^+ \in \overrightarrow{\beta^+} \cap \mathbf{fv}\, M$ and assume $[\mu]\gamma^+ \notin \overrightarrow{\alpha^+}$. Then since $T, \overrightarrow{\beta^+} \vdash \sigma_1 : \overrightarrow{\alpha^+}$, σ_1 acts as identity outside of $\overrightarrow{\alpha^+}$, i.e. $[\sigma_1][\mu]\gamma^+ = [\mu]\gamma^+$ (notice that γ^+ is in the domain of μ). Since $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \sigma_1 \simeq^{\leqslant} \mu^{-1} : \mathbf{fv}\, N$, application of σ_1 to $[\mu]\gamma^+ \in \mathbf{fv}\, N$ is equivalent to application of μ^{-1}, then $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\mu^{-1}][\mu]\gamma^+ \simeq^{\leqslant} [\mu]\gamma^+$, i.e. $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \gamma^+ \simeq^{\leqslant} [\mu]\gamma^+$, which means $\gamma^+ \in \mathbf{fv}\, [\mu]\gamma^+ \subseteq \mathbf{fv}\, N$. By assumption, $\gamma^+ \in \overrightarrow{\beta^+} \cap \mathbf{fv}\, M$, i.e. $\overrightarrow{\beta^+} \cap \mathbf{fv}\, N \neq \emptyset$, hence contradiction.
Second, we will show $T, \overrightarrow{\alpha^+} \vdash N \simeq^{\leqslant} [\mu|_{\overrightarrow{\beta^+}}]M$.
Since $T, \overrightarrow{\alpha^+} \vdash \sigma_2 : \overrightarrow{\beta^+}$ and $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \sigma_2 \simeq^{\leqslant} \mu : \mathbf{fv}\, M$, we have $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \sigma_2 \simeq^{\leqslant} \mu|_{\overrightarrow{\beta^+}} : \mathbf{fv}\, M$: for any $\alpha^{\pm} \in \mathbf{fv}\, M \setminus \overrightarrow{\beta^+}$, $[\sigma_2]\alpha^{\pm} = \alpha^{\pm}$ since $T, \overrightarrow{\alpha^+} \vdash \sigma_2 : \overrightarrow{\beta^+}$, and $[\mu|_{\overrightarrow{\beta^+}}]\alpha^{\pm} = \alpha^{\pm}$ by definition of substitution restriction; for $\beta^+ \in \overrightarrow{\beta^+}$, $[\mu|_{\overrightarrow{\beta^+}}]\beta^+ = [\mu]\beta^+$, and thus, $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\sigma_2]\beta^+ \simeq^{\leqslant} [\mu]\beta^+$ can be rewritten to $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\sigma_2]\beta^+ \simeq^{\leqslant} [\mu|_{\overrightarrow{\beta^+}}]\beta^+$.
By lemma 30, $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash \sigma_2 \simeq^{\leqslant} \mu|_{\overrightarrow{\beta^+}} : \mathbf{fv}\, M$ implies $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\sigma_2]M \simeq^{\leqslant} [\mu|_{\overrightarrow{\beta^+}}]M$. By similar reasoning, $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\sigma_1]N \simeq^{\leqslant} [\mu^{-1}|_{\overrightarrow{\alpha^+}}]N$.

This way, by transitivity of subtyping (lemma 22),

$$T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\mu^{-1}|_{\overrightarrow{\alpha^+}}]N \leqslant M \tag{5}$$

$$T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\mu|_{\overrightarrow{\beta^+}}]M \leqslant N \tag{6}$$

By applying $\mu|_{\overrightarrow{\beta^+}}$ to both sides of 5 (lemma 21), we have $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash [\mu|_{\overrightarrow{\beta^+}}][\mu^{-1}|_{\overrightarrow{\alpha^+}}]N \leqslant [\mu|_{\overrightarrow{\beta^+}}]M$. By contracting $\mu^{-1}|_{\overrightarrow{\alpha^+}} \circ \mu|_{\overrightarrow{\beta^+}} = \mu|^{-1}_{\overrightarrow{\beta^+}} \circ \mu|_{\overrightarrow{\beta^+}}$ (notice that $\mathbf{fv}\, N \cap \overrightarrow{\beta^+} = \emptyset$), we have $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash N \leqslant [\mu|_{\overrightarrow{\beta^+}}]M$, which together with 6 means $T, \overrightarrow{\alpha^+}, \overrightarrow{\beta^+} \vdash N \simeq^{\leqslant} [\mu|_{\overrightarrow{\beta^+}}]M$, and by strengthening, $T, \overrightarrow{\alpha^+} \vdash N \simeq^{\leqslant} [\mu|_{\overrightarrow{\beta^+}}]M$.

+ The proof is symmetric to the proof of the negative case.

□

Lemma 32 (Completeness of Equivalence). *Mutual subtyping implies declarative equivalence. Assuming all the types below are well-formed in T:*

+ *if $T \vdash P \simeq^{\leqslant} Q$ then $P \simeq^{D} Q$,*
− *if $T \vdash N \simeq^{\leqslant} M$ then $N \simeq^{D} M$.*

Proof. − Induction on the sum of sizes of N and M. By inversion, $T \vdash N \simeq^{\leqslant} M$ means $T \vdash N \leqslant M$ and $T \vdash M \leqslant N$. Let us consider the last rule that forms $T \vdash N \leqslant M$:

Case 1. (Var$^{\leqslant}_{-}$) i.e. $T \vdash N \leqslant M$ is of the form $T \vdash \alpha^- \leqslant \alpha^-$
Then $N \simeq^{D} M$ (i.e. $\alpha^- \simeq^{D} \alpha^-$) holds immediately by (Var$^{\simeq^D}_{-}$).

Case 2. ($\uparrow^{\leqslant}$) i.e. $T \vdash N \leqslant M$ is of the form $T \vdash {\uparrow}P \leqslant {\uparrow}Q$
Then by inversion, $T \vdash P \simeq^{\leqslant} Q$, and by induction hypothesis, $P \simeq^{D} Q$. Then $N \simeq^{D} M$ (i.e. ${\uparrow}P \simeq^{D} {\uparrow}Q$) holds by ($\uparrow^{\simeq^D}$).

Case 3. ($\rightarrow^{\leqslant}$) i.e. $T \vdash N \leqslant M$ is of the form $T \vdash P \rightarrow N' \leqslant Q \rightarrow M'$
Then by inversion, $T \vdash P \geqslant Q$ and $T \vdash N' \leqslant M'$. Notice that $T \vdash M \leqslant N$ is of the form $T \vdash Q \rightarrow M' \leqslant P \rightarrow N'$, which by inversion means $T \vdash Q \geqslant P$ and $T \vdash M' \leqslant N'$. This way, $T \vdash Q \simeq^{\leqslant} P$ and $T \vdash M' \simeq^{\leqslant} N'$. Then by induction hypothesis, $Q \simeq^{D} P$ and $M' \simeq^{D} N'$. Then $N \simeq^{D} M$ (i.e. $P \rightarrow N' \simeq^{D} Q \rightarrow M'$) holds by ($\rightarrow^{\simeq^D}$).

Case 4. ($\forall^{\leqslant}$) i.e. $T \vdash N \leqslant M$ is of the form $T \vdash \forall\overrightarrow{\alpha^+}.\, N' \leqslant \forall\overrightarrow{\beta^+}.\, M'$
Then by lemma 31, $T \vdash \forall\overrightarrow{\alpha^+}.\, N' \simeq^{\leqslant} \forall\overrightarrow{\beta^+}.\, M'$ means that there exists a bijection $\mu : \overrightarrow{\beta^+} \cap \mathbf{fv}\, M' \leftrightarrow \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N'$ such that $T, \overrightarrow{\alpha^+} \vdash [\mu]M' \simeq^{\leqslant} N'$.
Notice that the application of bijection μ to M' does not change its size (which is less than the size of M), hence the induction hypothesis applies. This way, $[\mu]M' \simeq^{D} N'$ (and by symmetry, $N' \simeq^{D} [\mu]M'$) holds by induction. Then we apply ($\forall^{\simeq^D}$) to get $\forall\overrightarrow{\alpha^+}.\, N' \simeq^{D} \forall\overrightarrow{\beta^+}.\, M'$, i.e. $N \simeq^{D} M$.

+ The proof is symmetric to the proof of the negative case.

□

8.5 Variable Ordering

Observation 1 (Ordering is deterministic). *If* **ord** *vars* **in** $N = \vec{\alpha}_1$ *and* **ord** *vars* **in** $N = \vec{\alpha}_2$ *then* $\vec{\alpha}_1 = \vec{\alpha}_2$. *If* **ord** *vars* **in** $P = \vec{\alpha}_1$ *and* **ord** *vars* **in** $P = \vec{\alpha}_2$ *then* $\vec{\alpha}_1 = \vec{\alpha}_2$. *This way, we can use* **ord** *vars* **in** N *and as a function on* N*, and* **ord** *vars* **in** P *as a function on* P.

Proof. By mutual structural induction on N and P. Notice that the shape of the term N or P uniquely determines the last used inference rule, and all the premises are deterministic on the input. □

Lemma 33 (Soundness of variable ordering). *Variable ordering extracts used free variables.*

- $-$ $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ N = \mathit{vars} \cap \textbf{fv}\ N$ *(as sets)*
- $+$ $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ P = \mathit{vars} \cap \textbf{fv}\ P$ *(as sets)*

Proof. We prove it by mutual induction on $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ N = \vec{\alpha}$ and $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ P = \vec{\alpha}$. The only non-trivial cases are $(\rightarrow^{\text{Ord}})$ and $(\forall^{\text{Ord}})$.

Case 1. $(\rightarrow^{\text{Ord}})$ Then the inferred ordering judgement has shape $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ P \rightarrow N = \vec{\alpha}_1, (\vec{\alpha}_2 \setminus \vec{\alpha}_1)$ and by inversion, $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ P = \vec{\alpha}_1$ and $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ N = \vec{\alpha}_2$.
By definition of free variables, $\mathit{vars} \cap \textbf{fv}\ P \rightarrow N = \mathit{vars} \cap \textbf{fv}\ P \cup \mathit{vars} \cap \textbf{fv}\ N$, and since by the induction hypothesis $\mathit{vars} \cap \textbf{fv}\ P = \vec{\alpha}_1$ and $\mathit{vars} \cap \textbf{fv}\ N = \vec{\alpha}_2$, we have $\mathit{vars} \cap \textbf{fv}\ P \rightarrow N = \vec{\alpha}_1 \cup \vec{\alpha}_2$.
On the other hand, As a set, $\vec{\alpha}_1 \cup \vec{\alpha}_2$ is equal to $\vec{\alpha}_1, (\vec{\alpha}_2 \setminus \vec{\alpha}_1)$.

Case 2. $(\forall^{\text{Ord}})$. Then the inferred ordering judgement has shape $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ \forall\overrightarrow{\alpha^+}.\ N = \vec{\alpha}$, and by inversion, $\mathit{vars} \cap \overrightarrow{\alpha^+} = \emptyset$ $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ N = \vec{\alpha}$. The latter implies that $\mathit{vars} \cap \textbf{fv}\ N = \vec{\alpha}$. We need to show that $\mathit{vars} \cap \textbf{fv}\ \forall\overrightarrow{\alpha^+}.\ N = \vec{\alpha}$, or equivalently, that $\mathit{vars} \cap (\textbf{fv}\ N \setminus \overrightarrow{\alpha^+}) = \mathit{vars} \cap \textbf{fv}\ N$, which holds since $\mathit{vars} \cap \overrightarrow{\alpha^+} = \emptyset$.

□

Corollary 12 (Additivity of ordering). *Variable ordering is additive (in terms of set union) with respect to its first argument.*

- $-$ $\textbf{ord}\ (\mathit{vars}_1 \cup \mathit{vars}_2)\ \textbf{in}\ N = \textbf{ord}\ \mathit{vars}_1\ \textbf{in}\ N \cup \textbf{ord}\ \mathit{vars}_2\ \textbf{in}\ N$ *(as sets)*
- $+$ $\textbf{ord}\ (\mathit{vars}_1 \cup \mathit{vars}_2)\ \textbf{in}\ P = \textbf{ord}\ \mathit{vars}_1\ \textbf{in}\ P \cup \textbf{ord}\ \mathit{vars}_2\ \textbf{in}\ P$ *(as sets)*

Lemma 34 (Weakening of ordering). *Only used variables matter in the first argument of the ordering,*

- $-$ $\textbf{ord}\ (\mathit{vars} \cap \textbf{fv}\ N)\ \textbf{in}\ N = \textbf{ord}\ \mathit{vars}\ \textbf{in}\ N$
- $+$ $\textbf{ord}\ (\mathit{vars} \cap \textbf{fv}\ P)\ \textbf{in}\ P = \textbf{ord}\ \mathit{vars}\ \textbf{in}\ P$

Proof. Mutual structural induction on N and P.

Case 1. If N is a variable α^-, we notice that $\alpha^- \in \mathit{vars}$ is equivalent to $\alpha^- \in \mathit{vars} \cap \alpha^-$.

Case 2. If N has shape $\uparrow P$, then the required property holds immediately by the induction hypothesis, since $\textbf{fv}\ (\uparrow P) = \textbf{fv}\ (P)$.

Case 3. If the term has shape $P \rightarrow N$ then $(\rightarrow^{\text{Ord}})$ was applied to infer $\textbf{ord}\ (\mathit{vars} \cap (\textbf{fv}\ P \cup \textbf{fv}\ N))\ \textbf{in}\ P \rightarrow N$ and $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ P \rightarrow N$. By inversion, the result of $\textbf{ord}\ (\mathit{vars} \cap (\textbf{fv}\ P \cup \textbf{fv}\ N))\ \textbf{in}\ P \rightarrow N$ depends on $A = \textbf{ord}\ (\mathit{vars} \cap (\textbf{fv}\ P \cup \textbf{fv}\ N))\ \textbf{in}\ P$ and $B = \textbf{ord}\ (\mathit{vars} \cap (\textbf{fv}\ P \cup \textbf{fv}\ N))\ \textbf{in}\ N$. The result of $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ P \rightarrow N$ depends on $X = \textbf{ord}\ \mathit{vars}\ \textbf{in}\ P$ and $Y = \textbf{ord}\ \mathit{vars}\ \textbf{in}\ N$.
Let us show that $A = B$ and $X = Y$, so the results are equal. By the induction hypothesis and set properties, $\textbf{ord}\ (\mathit{vars} \cap (\textbf{fv}\ P \cup \textbf{fv}\ N))\ \textbf{in}\ P = \textbf{ord}\ (\mathit{vars} \cap (\textbf{fv}\ P \cup \textbf{fv}\ N)) \cap \textbf{fv}\ (P)\ \textbf{in}\ P = \textbf{ord}\ \mathit{vars} \cap \textbf{fv}\ (P)\ \textbf{in}\ P = \textbf{ord}\ \mathit{vars}\ \textbf{in}\ P$. Analogously, $\textbf{ord}\ (\mathit{vars} \cap (\textbf{fv}\ P \cup \textbf{fv}\ N))\ \textbf{in}\ N = \textbf{ord}\ \mathit{vars}\ \textbf{in}\ N$.

Case 4. If the term has shape $\forall\overrightarrow{\alpha^+}.\ N$, we can assume that $\overrightarrow{\alpha^+}$ is disjoint from vars, since we operate on alpha-equivalence classes. Then using the induction hypothesis, set properties and $(\forall^{\text{Ord}})$: $\textbf{ord}\ \mathit{vars} \cap (\textbf{fv}\ (\forall\overrightarrow{\alpha^+}.\ N))\ \textbf{in}\ \forall\overrightarrow{\alpha^+}.\ N = \textbf{ord}\ \mathit{vars} \cap (\textbf{fv}\ (N) \setminus \overrightarrow{\alpha^+})\ \textbf{in}\ N = \textbf{ord}\ \mathit{vars} \cap (\textbf{fv}\ (N) \setminus \overrightarrow{\alpha^+}) \cap \textbf{fv}\ (N)\ \textbf{in}\ N = \textbf{ord}\ \mathit{vars} \cap \textbf{fv}\ (N)\ \textbf{in}\ N = \textbf{ord}\ \mathit{vars}\ \textbf{in}\ N$.

□

Corollary 13 (Idempotency of ordering).

- $-$ *If* $\textbf{ord}\ \mathit{vars}\ \textbf{in}\ N = \vec{\alpha}$ *then* $\textbf{ord}\ \vec{\alpha}\ \textbf{in}\ N = \vec{\alpha}$,

+ *If* **ord** *vars* **in** $P = \vec{\alpha}$ *then* **ord** $\vec{\alpha}$ **in** $P = \vec{\alpha}$*;*

Proof. By lemmas 33 and 34. □

Next, we make a set-theoretical observation that will be useful further. In general, any injective function (its image) distributes over the set intersection. However, for convenience, we allow the bijections on variables to be applied *outside of their domains* (as identities), which may violate the injectivity. To deal with these cases, we define a special notion of bijections collision-free on certain sets in such a way that a bijection that is collision-free on P and Q, distributes over intersection of P and Q.

Definition 29 (Collision-free bijection). *We say that a bijection $\mu : A \leftrightarrow B$ between sets of variables is **collision-free on sets** P and Q if and only if*

(1) $\mu(P \cap A) \cap Q = \emptyset$
(2) $\mu(Q \cap A) \cap P = \emptyset$

Observation 16. *Suppose that $\mu : A \leftrightarrow B$ is a bijection between two sets of variables, and μ is collision-free on P and Q. Then $\mu(P \cap Q) = \mu(P) \cap \mu(Q)$.*

Lemma 35 (Distributivity of renaming over variable ordering). *Suppose that μ is a bijection between two sets of variables $\mu : A \leftrightarrow B$.*

– *If μ is collision-free on vars and* **fv** N *then* $[\mu]($**ord** *vars* **in** $N) =$ **ord** $([\mu]$ *vars*$)$ **in** $[\mu]N$
+ *If μ is collision-free on vars and* **fv** P *then* $[\mu]($**ord** *vars* **in** $P) =$ **ord** $([\mu]$ *vars*$)$ **in** $[\mu]P$

Proof. Mutual induction on N and P.

Case 1. $N = \alpha^-$
let us consider four cases:

a. $\alpha^- \in A$ and $\alpha^- \in vars$
Then
$[\mu]($**ord** *vars* **in** $N) = [\mu]($**ord** *vars* **in** $\alpha^-)$
$= [\mu]\alpha^-$ by ($\text{Var}^{\text{Ord}}_{+\in}$)
$= \beta^-$ for some $\beta^- \in B$ (notice $\beta^- \in [\mu]vars$)
$=$ **ord** $[\mu]$ *vars* **in** β^- by ($\text{Var}^{\text{Ord}}_{+\in}$), because $\beta^- \in [\mu]vars$
$=$ **ord** $[\mu]$ *vars* **in** $[\mu]\alpha^-$

b. $\alpha^- \notin A$ and $\alpha^- \notin vars$
Notice that $[\mu]($**ord** *vars* **in** $N) = [\mu]($**ord** *vars* **in** $\alpha^-) = \cdot$ by ($\text{Var}^{\text{Ord}}_{+\notin}$). On the other hand, **ord** $[\mu]$ *vars* **in** $[\mu]\alpha^- =$ **ord** $[\mu]$ *vars* **in** $\alpha^- = \cdot$ The latter equality is from ($\text{Var}^{\text{Ord}}_{+\notin}$), because μ is collision-free on *vars* and **fv** N, so **fv** $N \ni \alpha^- \notin \mu(A \cap vars) \cup vars \supseteq [\mu]vars$.

c. $\alpha^- \in A$ but $\alpha^- \notin vars$
Then $[\mu]($**ord** *vars* **in** $N) = [\mu]($**ord** *vars* **in** $\alpha^-) = \cdot$ by ($\text{Var}^{\text{Ord}}_{+\notin}$). To prove that **ord** $[\mu]$ *vars* **in** $[\mu]\alpha^- = \cdot$, we apply ($\text{Var}^{\text{Ord}}_{+\notin}$). Let us show that $[\mu]\alpha^- \notin [\mu]vars$. Since $[\mu]\alpha^- = \mu(\alpha^-)$ and $[\mu]vars \subseteq \mu(A \cap vars) \cup vars$, it suffices to prove $\mu(\alpha^-) \notin \mu(A \cap vars) \cup vars$.

(i) If there is an element $x \in A \cap vars$ such that $\mu x = \mu\alpha^-$, then $x = \alpha^-$ by bijectivity of μ, which contradicts with $\alpha^- \notin vars$. This way, $\mu(\alpha^-) \notin \mu(A \cap vars)$.
(ii) Since μ is collision-free on *vars* and **fv** N, $\mu(A \cap$ **fv** $N) \ni \mu(\alpha^-) \notin vars$.

d. $\alpha^- \notin A$ but $\alpha^- \in vars$
ord $[\mu]$ *vars* **in** $[\mu]\alpha^- =$ **ord** $[\mu]$ *vars* **in** $\alpha^- = \alpha^-$. The latter is by ($\text{Var}^{\text{Ord}}_{+\notin}$), because

$\alpha^- = [\mu]\alpha^- \in [\mu]\mathit{vars}$ since $\alpha^- \in \mathit{vars}$. On the other hand, $[\mu](\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ N) = [\mu](\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ \alpha^-) = [\mu]\alpha^- = \alpha^-$.

Case 2. $N = \uparrow P$

$$\begin{aligned}[\mu](\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ N) &= [\mu](\mathbf{ord}\ \mathit{vars}\ \mathbf{in} \uparrow P)\\ &= [\mu](\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ P) && \text{by } (\uparrow^{\text{Ord}})\\ &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]P && \text{by the induction hypothesis}\\ &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in} \uparrow [\mu]P && \text{by } (\uparrow^{\text{Ord}})\\ &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]\uparrow P && \text{by the definition of substitution}\\ &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]N\end{aligned}$$

Case 3. $N = P \to M$

$$\begin{aligned}&[\mu](\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ N)\\ &= [\mu](\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ P \to M)\\ &= [\mu](\vec{\alpha}_1, (\vec{\alpha}_2 \setminus \vec{\alpha}_1)) && \text{where } \mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ P = \vec{\alpha}_1 \text{ and } \mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ M = \vec{\alpha}_2\\ &= [\mu]\vec{\alpha}_1, [\mu](\vec{\alpha}_2 \setminus \vec{\alpha}_1)\\ &= [\mu]\vec{\alpha}_1, ([\mu]\vec{\alpha}_2 \setminus [\mu]\vec{\alpha}_1) && \text{by induction on } \vec{\alpha}_2\text{; the inductive step is similar to case 1.}\\ &&& \text{Notice that } \mu \text{ is collision free on } \vec{\alpha}_1 \text{ and } \vec{\alpha}_2\\ &&& \text{since } \vec{\alpha}_1 \subseteq \mathit{vars} \text{ and } \vec{\alpha}_2 \subseteq \mathbf{fv}\ N\\ &= [\mu]\vec{\alpha}_1, ([\mu]\vec{\alpha}_2 \setminus [\mu]\vec{\alpha}_1)\end{aligned}$$

On the other hand,

$$\begin{aligned}&\mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]N\\ &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]P \to [\mu]M\\ &= \vec{\beta}_1, (\vec{\beta}_2 \setminus \vec{\beta}_1) && \text{where } \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]P = \vec{\beta}_1 \text{ and}\\ &&& \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]M = \vec{\beta}_2\text{; then by the induction}\\ &&& \text{hypothesis, } \vec{\beta}_1 = [\mu]\vec{\alpha}_1,\ \vec{\beta}_2 = [\mu]\vec{\alpha}_2\\ &= [\mu]\vec{\alpha}_1, ([\mu]\vec{\alpha}_2 \setminus [\mu]\vec{\alpha}_1)\end{aligned}$$

Case 4. $N = \forall\overrightarrow{\alpha^+}.\ M$

$$\begin{aligned}[\mu](\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ N) &= [\mu]\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ \forall\overrightarrow{\alpha^+}.\ M\\ &= [\mu]\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ M\\ &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]M && \text{by the induction hypothesis}\end{aligned}$$

$$\begin{aligned}(\mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]N) &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]\forall\overrightarrow{\alpha^+}.\ M\\ &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ \forall\overrightarrow{\alpha^+}.\ [\mu]M\\ &= \mathbf{ord}\ [\mu]\mathit{vars}\ \mathbf{in}\ [\mu]M\end{aligned}$$

□

Lemma 36 (Ordering is not affected by independent substitutions). *Suppose that $T_2 \vdash \sigma : T_1$, i.e. σ maps variables from T_1 into types taking free variables from T_2, and vars is a set of variables disjoint with both T_1 and T_2, N and P are types. Then*

- $-$ $\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ [\sigma]N = \mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ N$
- $+$ $\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ [\sigma]P = \mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ P$

Proof. Mutual induction on N and P.

Case 1. $N = \alpha^-$
If $\alpha^- \notin T_1$ then $[\sigma]\alpha^- = \alpha^-$ and $\mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]\alpha^- = \mathbf{ord}\ vars\ \mathbf{in}\ \alpha^-$, as requried. If $\alpha^- \in T_1$ then $\alpha^- \notin vars$, so $\mathbf{ord}\ vars\ \mathbf{in}\ \alpha^- = \cdot$. Moreover, $T_2 \vdash \sigma : T_1$ means $\mathbf{fv}\,([\sigma]\alpha^-) \subseteq T_2$, and thus, as a set, $\mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]\alpha^- = vars \cap \mathbf{fv}\,([\sigma]\alpha^-) \subseteq vars \cap T_2 = \cdot$.

Case 2. $N = \forall\overrightarrow{\alpha^+}.\,M$
We can assume $\overrightarrow{\alpha^+} \cap T_1 = \emptyset$ and $\overrightarrow{\alpha^+} \cap vars = \emptyset$. Then

$$\begin{aligned}
\mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]N &= \mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]\forall\overrightarrow{\alpha^+}.\,M\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ \forall\overrightarrow{\alpha^+}.\,[\sigma]M\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]M && \text{by the induction hypothesis}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ M\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ \forall\overrightarrow{\alpha^+}.\,M\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ N
\end{aligned}$$

Case 3. $N = \uparrow P$

$$\begin{aligned}
\mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]N &= \mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]\uparrow P\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ \uparrow[\sigma]P && \text{by the definition of substitution}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]P && \text{by the induction hypothesis}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ P && \text{by the definition of substitution}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ \uparrow P && \text{by the definition of ordering}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ N
\end{aligned}$$

Case 4. $N = P \to M$

$$\begin{aligned}
\mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]N &= \mathbf{ord}\ vars\ \mathbf{in}\ [\sigma](P \to M)\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ ([\sigma]P \to [\sigma]M) && \text{def. of substitution}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]P,\\
&\quad (\mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]M \setminus \mathbf{ord}\ vars\ \mathbf{in}\ [\sigma]P) && \text{def. of ordering}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ P,\\
&\quad (\mathbf{ord}\ vars\ \mathbf{in}\ M \setminus \mathbf{ord}\ vars\ \mathbf{in}\ P) && \text{the induction hypothesis}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ P \to M && \text{def. of ordering}\\
&= \mathbf{ord}\ vars\ \mathbf{in}\ N
\end{aligned}$$

Case 5. The proofs of the positive cases are symmetric.

□

Lemma 37 (Completeness of variable ordering). *Variable ordering is invariant under equivalence. For arbitrary vars,*

– *If* $N \simeq^D M$ *then* $\mathbf{ord}\ vars\ \mathbf{in}\ N = \mathbf{ord}\ vars\ \mathbf{in}\ M$ *(as lists)*
+ *If* $P \simeq^D Q$ *then* $\mathbf{ord}\ vars\ \mathbf{in}\ P = \mathbf{ord}\ vars\ \mathbf{in}\ Q$ *(as lists)*

Proof. Mutual induction on $N \simeq^D M$ and $P \simeq^D Q$. Let us consider the rule inferring $N \simeq^D M$.

Case 1. $(\textsc{Var}_-^{\simeq^D})$

Case 2. $(\uparrow^{\simeq^D})$

Case 3. $(\to^{\simeq^D})$ Then the equivalence has shape $P \to N \simeq^D Q \to M$, and by inversion, $P \simeq^D Q$ and $N \simeq^D M$. Then by the induction hypothesis, $\mathbf{ord}\ vars\ \mathbf{in}\ P = \mathbf{ord}\ vars\ \mathbf{in}\ Q$ and $\mathbf{ord}\ vars\ \mathbf{in}\ N = \mathbf{ord}\ vars\ \mathbf{in}\ M$. Since the resulting ordering for $P \to N$ and $Q \to M$ depend on the ordering of the corresponding components, which are equal, the results are equal.

Case 4. $(\forall^{\simeq^D})$ Then the equivalence has shape $\forall\overrightarrow{\alpha^+}.\ N \simeq^D \forall\overrightarrow{\beta^+}.\ M$. and by inversion there exists $\mu : (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M) \leftrightarrow (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N)$ such that

- $\overrightarrow{\alpha^+} \cap \mathbf{fv}\, M = \emptyset$ and
- $N \simeq^D [\mu]M$

Let us assume that *vars* is disjoint from $\overrightarrow{\alpha^+}$ and $\overrightarrow{\beta^+}$ (we can always alpha-rename the bound variables). Then $\mathbf{ord}\ vars\ \mathbf{in}\ \forall\overrightarrow{\alpha^+}.\ N = \mathbf{ord}\ vars\ \mathbf{in}\ N$, $\mathbf{ord}\ vars\ \mathbf{in}\ \forall\overrightarrow{\alpha^+}.\ M = \mathbf{ord}\ vars\ \mathbf{in}\ M$ and by the induction hypothesis, $\mathbf{ord}\ vars\ \mathbf{in}\ N = \mathbf{ord}\ vars\ \mathbf{in}\ [\mu]M$. This way, it suffices tho show that $\mathbf{ord}\ vars\ \mathbf{in}\ [\mu]M = \mathbf{ord}\ vars\ \mathbf{in}\ M$. It holds by lemma 36 since *vars* is disjoint form the domain and the codomain of $\mu : (\overrightarrow{\beta^+} \cap \mathbf{fv}\, M) \leftrightarrow (\overrightarrow{\alpha^+} \cap \mathbf{fv}\, N)$ by assumption.

Case 5. The positive cases are proved symmetrically.

□

8.6 Normaliztaion

Observation 2 (Normalization is deterministic). *If* $\mathbf{nf}\,(N) = M$ *and* $\mathbf{nf}\,(N) = M'$ *then* $M = M'$. *If* $\mathbf{nf}\,(P) = Q$ *and* $\mathbf{nf}\,(P) = Q'$ *then* $Q = Q'$. *This way, we can use normalization as a function.*

Proof. By straightforward induction using observation 1. □

Lemma 38. *Free variables are not changed by the normalization*

$-$ $\mathbf{fv}\, N = \mathbf{fv}\,\mathbf{nf}\,(N)$
$+$ $\mathbf{fv}\, P = \mathbf{fv}\,\mathbf{nf}\,(P)$

Proof. By mutual induction on N and P. The base cases ((Var$^{\text{NF}}_{-}$) and (Var$^{\text{NF}}_{+}$)) are trivial; the congruent cases (($\uparrow^{\text{NF}}$), ($\downarrow^{\text{NF}}$), and ($\rightarrow^{\text{NF}}$)) are proved by the induction hypothesis.

Let us consider the case when the term is formed by $\forall$, that is the normalization judgment has a shape $\mathbf{nf}\,(\forall\overrightarrow{\alpha^+}.\ N) = \forall\overrightarrow{\alpha^+}'.\ N'$, where by inversion $\mathbf{nf}\,(N) = N'$ and $\mathbf{ord}\ \overrightarrow{\alpha^+}\ \mathbf{in}\ N' = \overrightarrow{\alpha^+}'$. By the induction hypothesis, $\mathbf{fv}\, N = \mathbf{fv}\, N'$. Since $\mathbf{fv}\,(\forall\overrightarrow{\alpha^+}.\ N) = \mathbf{fv}\, N \setminus \overrightarrow{\alpha^+}$, and $\mathbf{fv}\,(\forall\overrightarrow{\alpha^+}'.\ N') = \mathbf{fv}\, N' \setminus \overrightarrow{\alpha^+}'$, it is left to show that $\mathbf{fv}\, N \setminus \overrightarrow{\alpha^+} = \mathbf{fv}\, N \setminus \overrightarrow{\alpha^+}'$. By lemma 37, $\overrightarrow{\alpha^+}' = \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N' = \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N$. Then $\mathbf{fv}\, N \setminus \overrightarrow{\alpha^+} = \mathbf{fv}\, N \setminus (\overrightarrow{\alpha^+} \cup \mathbf{fv}\, N)$ by set-theoretic properties, and thus, $\mathbf{fv}\, N \setminus \overrightarrow{\alpha^+} = \mathbf{fv}\, N \setminus \overrightarrow{\alpha^+}'$.

The case when the term is positive and formed by $\exists$ is symmetric. □

Lemma 39 (Soundness of normalization).

$-$ $N \simeq^D \mathbf{nf}\,(N)$
$+$ $P \simeq^D \mathbf{nf}\,(P)$

Proof. Mutual induction on $\mathbf{nf}\,(N) = M$ and $\mathbf{nf}\,(P) = Q$. Let us consider how this judgment is formed:

Case 1. (Var$^{\text{NF}}_{-}$) and (Var$^{\text{NF}}_{+}$)
By the corresponding equivalence rules.

Case 2. ($\uparrow^{\text{NF}}$), ($\downarrow^{\text{NF}}$), and ($\rightarrow^{\text{NF}}$)
By the induction hypothesis and the corresponding congruent equivalence rules.

Case 3. ($\forall^{\text{NF}}$), i.e. $\mathbf{nf}\,(\forall\overrightarrow{\alpha^+}.\ N) = \forall\overrightarrow{\alpha^+}'.\ N'$
From the induction hypothesis, we know that $N \simeq^D N'$. In particular, by lemma 24, $\mathbf{fv}\, N \equiv \mathbf{fv}\, N'$. Then by lemma 33, $\overrightarrow{\alpha^+}' \equiv \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N' \equiv \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N$, and thus, $\overrightarrow{\alpha^+}' \cap \mathbf{fv}\, N' \equiv \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N$. To prove $\forall\overrightarrow{\alpha^+}.\ N \simeq^D \forall\overrightarrow{\alpha^+}'.\ N'$, it suffices to provide a bijection $\mu : \overrightarrow{\alpha^+}' \cap \mathbf{fv}\, N' \leftrightarrow \overrightarrow{\alpha^+} \cap \mathbf{fv}\, N$ such that $N \simeq^D [\mu]N'$. Since these sets are equal, we take $\mu = id$.

Case 4. ($\exists^{\text{NF}}$) Same as for case 3.

□

Corollary 14 (Normalization preserves well-formedness).

+ $T \vdash P \iff T \vdash \mathbf{nf}\,(P)$,
− $T \vdash N \iff T \vdash \mathbf{nf}\,(N)$

Proof. Immediately from lemmas 26 and 39. □

Corollary 15 (Normalization preserves well-formedness of substitution). $T_2 \vdash \sigma : T_1 \iff T_2 \vdash \mathbf{nf}\,(\sigma) : T_1$

Proof. Let us prove the forward direction. Suppose that $\alpha^\pm \in T_1$. Let us show that $T_2 \vdash [\mathbf{nf}\,(\sigma)]\alpha^\pm$. By the definition of substitution normalization, $[\mathbf{nf}\,(\sigma)]\alpha^\pm = \mathbf{nf}\,([\sigma]\alpha^\pm)$. Then by corollary 14, to show $T_2 \vdash \mathbf{nf}\,([\sigma]\alpha^\pm)$, it suffices to show $T_2 \vdash [\sigma]\alpha^\pm$, which holds by the assumption $T_2 \vdash \sigma : T_1$.

The backward direction is proved analogously. □

Lemma 40 (Normalization preserves substitution signature). *Suppose that σ is a substitution, T_1 and T_2 are contexts. Then $T_2 \vdash \sigma : T_1$ implies $T_2 \vdash \mathbf{nf}\,(\sigma) : T_1$.*

Proof. Suppose that $\alpha^\pm \in T_1$. Then by corollary 14, $T_2 \vdash \mathbf{nf}\,([\sigma]\alpha^\pm) = [\mathbf{nf}\,(\sigma)]\alpha^\pm$ is equivalent to $T_2 \vdash [\sigma]\alpha^\pm$.

Suppose that $\alpha^\pm \notin T_1$. $T_2 \vdash \sigma : T_1$ means that $[\sigma]\alpha^\pm = \alpha^\pm$, and then $[\mathbf{nf}\,(\sigma)]\alpha^\pm = \mathbf{nf}\,([\sigma]\alpha^\pm) = \mathbf{nf}\,(\alpha^\pm) = \alpha^\pm$. □

Corollary 16 (Normalization is sound w.r.t. subtyping-induced equivalence).

+ *if $T \vdash P$ then $T \vdash P \simeq^{\leqslant} \mathbf{nf}\,(P)$,*
− *if $T \vdash N$ then $T \vdash N \simeq^{\leqslant} \mathbf{nf}\,(N)$.*

Proof. Immediately from lemmas 27 and 39 and corollary 14. □

Corollary 17 (Normalization preserves subtyping). *Assuming all the types are well-formed in context T,*

+ $T \vdash P \geqslant Q \iff T \vdash \mathbf{nf}\,(P) \geqslant \mathbf{nf}\,(Q)$,
− $T \vdash N \leqslant M \iff T \vdash \mathbf{nf}\,(N) \leqslant \mathbf{nf}\,(M)$.

Proof.

+ $\Rightarrow$ Let us assume $T \vdash P \geqslant Q$. By corollary 16, $T \vdash P \simeq^{\leqslant} \mathbf{nf}\,(P)$ and $T \vdash Q \simeq^{\leqslant} \mathbf{nf}\,(Q)$, in particular, by inversion, $T \vdash \mathbf{nf}\,(P) \geqslant P$ and $T \vdash Q \geqslant \mathbf{nf}\,(Q)$. Then by transitivity of subtyping (lemma 22), $T \vdash \mathbf{nf}\,(P) \geqslant \mathbf{nf}\,(Q)$.
  $\Leftarrow$ Let us assume $T \vdash \mathbf{nf}\,(P) \geqslant \mathbf{nf}\,(Q)$. Also by corollary 16 and inversion, $T \vdash P \geqslant \mathbf{nf}\,(P)$ and $T \vdash \mathbf{nf}\,(Q) \geqslant Q$. Then by the transitivity, $T \vdash P \geqslant Q$.
− The negative case is proved symmetrically.

□

Corollary 18 (Normalization preserves ordering). *For any vars,*

− $\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ \mathbf{nf}\,(N) = \mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ M$
+ $\mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ \mathbf{nf}\,(P) = \mathbf{ord}\ \mathit{vars}\ \mathbf{in}\ Q$

Proof. Immediately from lemmas 37 and 39. □

Lemma 41 (Distributivity of normalization over substitution). *Normalization of a term distributes over substitution. Suppose that σ is a substitution, N and P are types. Then*

− $\mathbf{nf}\,([\sigma]N) = [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(N)$

+ $\mathbf{nf}\,([\sigma]P) = [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(P)$

where $\mathbf{nf}\,(\sigma)$ *means pointwise normalization:* $[\mathbf{nf}\,(\sigma)]\alpha^- = \mathbf{nf}\,([\sigma]\alpha^-)$.

Proof. Mutual induction on N and P.

Case 1. $N = \alpha^-$

$\mathbf{nf}\,([\sigma]N) = \mathbf{nf}\,([\sigma]\alpha^-) = [\mathbf{nf}\,(\sigma)]\alpha^-$.

$[\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(N) = [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(\alpha^-) = [\mathbf{nf}\,(\sigma)]\alpha^-$.

Case 2. $P = \alpha^+$

Similar to case 1.

Case 3. If the type is formed by $\rightarrow$, $\uparrow$, or $\downarrow$, the required equality follows from the congruence of the normalization and substitution and the induction hypothesis. For example, if $N = P \rightarrow M$ then

$$\begin{aligned}
\mathbf{nf}\,([\sigma]N) &= \mathbf{nf}\,([\sigma](P \rightarrow M)) \\
&= \mathbf{nf}\,([\sigma]P \rightarrow [\sigma]M) && \text{By congruence of substitution} \\
&= \mathbf{nf}\,([\sigma]P) \rightarrow \mathbf{nf}\,([\sigma]M) && \text{By congruence of normalization} \\
&= [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(P) \rightarrow [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(M) && \text{By induction hypothesis} \\
&= [\mathbf{nf}\,(\sigma)](\mathbf{nf}\,(P) \rightarrow \mathbf{nf}\,(M)) && \text{By congruence of substitution} \\
&= [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(P \rightarrow M) && \text{By congruence of normalization} \\
&= [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(N)
\end{aligned}$$

Case 4. $N = \forall\overrightarrow{\alpha^+}.\,M$

$$\begin{aligned}
[\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(N) &= [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(\forall\overrightarrow{\alpha^+}.\,M) \\
&= [\mathbf{nf}\,(\sigma)]\forall\overrightarrow{\alpha^+}{}'.\,\mathbf{nf}\,(M) && \text{Where } \overrightarrow{\alpha^+}{}' = \mathbf{ord}\,\overrightarrow{\alpha^+}\,\mathbf{in}\,\mathbf{nf}\,(M) = \mathbf{ord}\,\overrightarrow{\alpha^+}\,\mathbf{in}\,M \\
& && \text{(the latter is by corollary 18)}
\end{aligned}$$

$$\begin{aligned}
\mathbf{nf}\,([\sigma]N) &= \mathbf{nf}\,([\sigma]\forall\overrightarrow{\alpha^+}.\,M) \\
&= \mathbf{nf}\,(\forall\overrightarrow{\alpha^+}.\,[\sigma]M) && \text{Assuming } \overrightarrow{\alpha^+} \cap T_1 = \emptyset \text{ and } \overrightarrow{\alpha^+} \cap T_2 = \emptyset \\
&= \forall\overrightarrow{\beta^+}.\,\mathbf{nf}\,([\sigma]M) && \text{Where } \overrightarrow{\beta^+} = \mathbf{ord}\,\overrightarrow{\alpha^+}\,\mathbf{in}\,\mathbf{nf}\,([\sigma]M) = \mathbf{ord}\,\overrightarrow{\alpha^+}\,\mathbf{in}\,[\sigma]M \\
& && \text{(the latter is by corollary 18)} \\
&= \forall\overrightarrow{\alpha^+}{}'.\,\mathbf{nf}\,([\sigma]M) && \text{By lemma 36,} \\
& && \overrightarrow{\beta^+} = \overrightarrow{\alpha^+}{}' \text{ since } \overrightarrow{\alpha^+} \text{ is disjoint with } T_1 \text{ and } T_2 \\
&= \forall\overrightarrow{\alpha^+}{}'.\,[\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(M) && \text{By the induction hypothesis}
\end{aligned}$$

To show the alpha-equivalence of $[\mathbf{nf}\,(\sigma)]\forall\overrightarrow{\alpha^+}{}'.\,\mathbf{nf}\,(M)$ and $\forall\overrightarrow{\alpha^+}{}'.\,[\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(M)$, we can assume that $\overrightarrow{\alpha^+}{}' \cap T_1 = \emptyset$, and $\overrightarrow{\alpha^+}{}' \cap T_2 = \emptyset$.

Case 5. $P = \exists\overrightarrow{\alpha^-}.\,Q$

Same as for case 4.

□

Corollary 19 (Commutativity of normalization and renaming)**.** *Normalization of a term commutes with renaming. Suppose that* μ *is a bijection between two sets of variables* $\mu : A \leftrightarrow B$*. Then*

− $\mathbf{nf}\,([\mu]N) = [\mu]\mathbf{nf}\,(N)$

+ $\mathbf{nf}\,([\mu]P) = [\mu]\mathbf{nf}\,(P)$

Proof. Immediately from lemma 41, after noticing that $\mathbf{nf}\,(\mu) = \mu$. □

Lemma 42 (Completeness of Normalization w.r.t. Declarative Equivalence)**.** *Normalization returns the same representative for equivalent types.*

– *If* $N \simeq^D M$ *then* $\mathbf{nf}\,(N) = \mathbf{nf}\,(M)$,
+ *if* $P \simeq^D Q$ *then* $\mathbf{nf}\,(P) = \mathbf{nf}\,(Q)$.

Proof. Mutual induction on $N \simeq^D M$ and $P \simeq^D Q$.

Case 1. $(\forall^{\simeq^D})$

From the definition of the normalization,

- $\mathbf{nf}\,(\forall\overrightarrow{\alpha^+}.\,N) = \forall\overrightarrow{\alpha^+}'.\,\mathbf{nf}\,(N)$ where $\overrightarrow{\alpha^+}'$ is $\mathbf{ord}\,\overrightarrow{\alpha^+}\,\mathbf{in}\,\mathbf{nf}\,(N)$
- $\mathbf{nf}\,(\forall\overrightarrow{\beta^+}.\,M) = \forall\overrightarrow{\beta^+}'.\,\mathbf{nf}\,(M)$ where $\overrightarrow{\beta^+}'$ is $\mathbf{ord}\,\overrightarrow{\beta^+}\,\mathbf{in}\,\mathbf{nf}\,(M)$

Let us take $\mu : (\overrightarrow{\beta^+} \cap \mathbf{fv}\,M) \leftrightarrow (\overrightarrow{\alpha^+} \cap \mathbf{fv}\,N)$ from the inversion of the equivalence judgment. Notice that from lemmas 33 and 38, the domain and the codomain of μ can be written as $\mu : \overrightarrow{\beta^+}' \leftrightarrow \overrightarrow{\alpha^+}'$.

To show the alpha-equivalence of $\forall\overrightarrow{\alpha^+}'.\,\mathbf{nf}\,(N)$ and $\forall\overrightarrow{\beta^+}'.\,\mathbf{nf}\,(M)$, it suffices to prove that (i) $[\mu]\mathbf{nf}\,(M) = \mathbf{nf}\,(N)$ and (ii) $[\mu]\overrightarrow{\beta^+}' = \overrightarrow{\alpha^+}'$.

(i) $[\mu]\mathbf{nf}\,(M) = \mathbf{nf}\,([\mu]M) = \mathbf{nf}\,(N)$. The first equality holds by corollary 19, the second—by the induction hypothesis.

(ii)
$$
\begin{aligned}
[\mu]\overrightarrow{\beta^+}' &= [\mu]\mathbf{ord}\,\overrightarrow{\beta^+}\,\mathbf{in}\,\mathbf{nf}\,(M) && \text{by the definition of } \overrightarrow{\beta^+}' \\
&= [\mu]\mathbf{ord}\,(\overrightarrow{\beta^+} \cap \mathbf{fv}\,M)\,\mathbf{in}\,\mathbf{nf}\,(M) && \text{from lemmas 34 and 38} \\
&= \mathbf{ord}\,[\mu](\overrightarrow{\beta^+} \cap \mathbf{fv}\,M)\,\mathbf{in}\,[\mu]\mathbf{nf}\,(M) && \text{by lemma 35, because} \\
& && \overrightarrow{\alpha^+} \cap \mathbf{fv}\,N \cap \mathbf{fv}\,\mathbf{nf}\,(M) \subseteq \overrightarrow{\alpha^+} \cap \mathbf{fv}\,M = \emptyset \\
& && \overrightarrow{\alpha^+} \cap \mathbf{fv}\,N \cap (\overrightarrow{\beta^+} \cap \mathbf{fv}\,M) \subseteq \overrightarrow{\alpha^+} \cap \mathbf{fv}\,M = \emptyset \\
&= \mathbf{ord}\,[\mu](\overrightarrow{\beta^+} \cap \mathbf{fv}\,M)\,\mathbf{in}\,\mathbf{nf}\,(N) && \text{since } [\mu]\mathbf{nf}\,(M) = \mathbf{nf}\,(N) \text{ is proved} \\
&= \mathbf{ord}\,(\overrightarrow{\alpha^+} \cap \mathbf{fv}\,N)\,\mathbf{in}\,\mathbf{nf}\,(N) && \text{because } \mu \text{ is a bijection between} \\
& && \overrightarrow{\alpha^+} \cap \mathbf{fv}\,N \text{ and } \overrightarrow{\beta^+} \cap \mathbf{fv}\,M \\
&= \mathbf{ord}\,\overrightarrow{\alpha^+}\,\mathbf{in}\,\mathbf{nf}\,(N) && \text{from lemmas 34 and 38} \\
&= \overrightarrow{\alpha^+}' && \text{by the definition of } \overrightarrow{\alpha^+}'
\end{aligned}
$$

Case 2. $(\exists^{\simeq^D})$ Same as for case 1.

Case 3. Other rules are congruent, and thus, proved by the corresponding congruent alpha-equivalence rule, which is applicable by the induction hypothesis.

□

Lemma 43 (Algorithmization of Declarative Equivalence)**.** *Declarative equivalence is the equality of normal forms.*

+ $P \simeq^D Q \iff \mathbf{nf}\,(P) = \mathbf{nf}\,(Q)$,
– $N \simeq^D M \iff \mathbf{nf}\,(N) = \mathbf{nf}\,(M)$.

Proof.

+ Let us prove both directions separately.
  - $\Rightarrow$ exactly by lemma 42,

$\Leftarrow$ from lemma 39, we know $P \simeq^D \mathbf{nf}(P) = \mathbf{nf}(Q) \simeq^D Q$, then by transitivity (lemma 25), $P \simeq^D Q$.

– For the negative case, the proof is the same.

□

Corollary 20 (Completeness of Normalization w.r.t. Subtyping-Induced Equivalence). *Assuming all the types below are well-formed in T:*

+ *if* $T \vdash P \simeq^{\leqslant} Q$ *then* $\mathbf{nf}(P) = \mathbf{nf}(Q)$,
– *if* $T \vdash N \simeq^{\leqslant} M$ *then* $\mathbf{nf}(N) = \mathbf{nf}(M)$.

Proof. Immediately from lemmas 32 and 42. □

Lemma 44 (Idempotence of normalization). *Normalization is idempotent*

– $\mathbf{nf}(\mathbf{nf}(N)) = \mathbf{nf}(N)$
+ $\mathbf{nf}(\mathbf{nf}(P)) = \mathbf{nf}(P)$

Proof. By applying lemma 42 to lemma 39. □

Lemma 45. *The result of a substitution is normalized if and only if the initial type and the substitution are normalized.*

Suppose that σ is a substitution $T_2 \vdash \sigma : T_1$, P is a positive type ($T_1 \vdash P$), N is a negative type ($T_1 \vdash N$). Then

$$+ \ [\sigma]P \text{ is normal} \iff \begin{cases} \sigma|_{\mathbf{fv}(P)} & \text{is normal} \\ P & \text{is normal} \end{cases}$$

$$- \ [\sigma]N \text{ is normal} \iff \begin{cases} \sigma|_{\mathbf{fv}(N)} & \text{is normal} \\ N & \text{is normal} \end{cases}$$

Proof. Mutual induction on $T_1 \vdash P$ and $T_1 \vdash N$.

Case 1. $N = \alpha^-$
Then N is always normal, and the normality of $\sigma|_{\alpha^-}$ by the definition means $[\sigma]\alpha^-$ is normal.

Case 2. $N = P \to M$

$$\begin{aligned}
[\sigma](P \to M) \text{ is normal} &\iff [\sigma]P \to [\sigma]M \text{ is normal} && \text{by substitution congruence} \\
&\iff \begin{cases} [\sigma]P & \text{is normal} \\ [\sigma]M & \text{is normal} \end{cases} \\
&\iff \begin{cases} P & \text{is normal} \\ \sigma|_{\mathbf{fv}(P)} & \text{is normal} \\ M & \text{is normal} \\ \sigma|_{\mathbf{fv}(M)} & \text{is normal} \end{cases} && \text{by the induction hypothesis} \\
&\iff \begin{cases} P \to M & \text{is normal} \\ \sigma|_{\mathbf{fv}(P) \cup \mathbf{fv}(M)} & \text{is normal} \end{cases} \\
&\iff \begin{cases} P \to M & \text{is normal} \\ \sigma|_{\mathbf{fv}(P \to M)} & \text{is normal} \end{cases}
\end{aligned}$$

Case 3. $N = \uparrow P$
By congruence and the inductive hypothesis, similar to case 2

Case 4. $N = \forall\overrightarrow{\alpha^+}.\ M$
$[\sigma](\forall\overrightarrow{\alpha^+}.\ M)$ is normal

$$\begin{aligned}
&\iff (\forall\overrightarrow{\alpha^+}.\ [\sigma]M) \text{ is normal} && \text{assuming } \overrightarrow{\alpha^+} \cap T_1 = \emptyset \text{ and } \overrightarrow{\alpha^+} \cap T_2 = \emptyset\\
&\iff \begin{cases} [\sigma]M \text{ is normal} \\ \textbf{ord}\ \overrightarrow{\alpha^+}\ \textbf{in}\ [\sigma]M = \overrightarrow{\alpha^+} \end{cases} && \text{by the definition of normalization}\\
&\iff \begin{cases} [\sigma]M \text{ is normal} \\ \textbf{ord}\ \overrightarrow{\alpha^+}\ \textbf{in}\ M = \overrightarrow{\alpha^+} \end{cases} && \text{by lemma 36}\\
&\iff \begin{cases} \sigma|_{\mathbf{fv}\,(M)} \text{ is normal} \\ M \text{ is normal} \\ \textbf{ord}\ \overrightarrow{\alpha^+}\ \textbf{in}\ M = \overrightarrow{\alpha^+} \end{cases} && \text{by the induction hypothesis}\\
&\iff \begin{cases} \sigma|_{\mathbf{fv}\,(\forall\overrightarrow{\alpha^+}.\ M)} \text{ is normal} \\ \forall\overrightarrow{\alpha^+}.\ M \text{ is normal} \end{cases} && \begin{array}{l}\text{since } \mathbf{fv}\,(\forall\overrightarrow{\alpha^+}.\ M) = \mathbf{fv}\,(M);\\ \text{by the definition of normalization}\end{array}
\end{aligned}$$

Case 5. $P = \ldots$
The positive cases are done in the same way as the negative ones.

□

Lemma 46 (Algorithmization of subtyping-induced equivalence). *Mutual subtyping is the equality of normal forms. Assuming all the types below are well-formed in T:*

+ $T \vdash P \simeq^{\leqslant} Q \iff \mathbf{nf}\,(P) = \mathbf{nf}\,(Q)$,
− $T \vdash N \simeq^{\leqslant} M \iff \mathbf{nf}\,(N) = \mathbf{nf}\,(M)$.

Proof. Let us prove the positive case, the negative case is symmetric. We prove both directions of $\iff$ separately:

$\Rightarrow$ exactly corollary 20;
$\Leftarrow$ by lemmas 27 and 43.

□

Corollary 21 (Substitution preserves declarative equivalence). *Suppose that σ is a substitution. Then*

+ $P \simeq^D Q$ *implies* $[\sigma]P \simeq^D [\sigma]Q$
− $N \simeq^D M$ *implies* $[\sigma]N \simeq^D [\sigma]M$

Proof.
$$\begin{aligned}
P \simeq^D Q &\Rightarrow \mathbf{nf}\,(P) = \mathbf{nf}\,(Q) && \text{by lemma 46}\\
&\Rightarrow [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(P) = [\mathbf{nf}\,(\sigma)]\mathbf{nf}\,(Q)\\
&\Rightarrow \mathbf{nf}\,([\sigma]P) = \mathbf{nf}\,([\sigma]Q) && \text{by lemma 41}\\
&\Rightarrow [\sigma]P \simeq^D [\sigma]Q && \text{by lemma 46}
\end{aligned}$$
□

9 PROPERTIES OF THE ALGORITHMIC TYPE SYSTEM

9.1 Algorithmic Type Well-formedness

Lemma 50 (Soundness of algorithmic type well-formedness).

+ *if* $T\ ;\Upsilon \vdash P$ *then* $\mathbf{fv}\,(P) \subseteq T$ *and* $\mathbf{fav}(P) \subseteq \Upsilon$*;*
− *if* $T\ ;\Upsilon \vdash N$ *then* $\mathbf{fv}\,(N) \subseteq T$ *and* $\mathbf{fav}(N) \subseteq \Upsilon$*.*

Proof. The proof is analogous to lemma 1. The additional base case is when $T ; \Upsilon \vdash P$ is derived by (UVar$^{\text{WF}}_{+}$), and the symmetric negative case. In this case, $P = \widehat{\alpha}^{+}$, and $\mathbf{fav}(P) = \widehat{\alpha}^{+} \subseteq \Upsilon$ by inversion; $\mathbf{fv}\,(P) = \emptyset \subseteq T$ vacuously. □

Lemma 51 (Completeness of algorithmic type well-formedness). *In the well-formedness judgment, only used variables matter:*

+ *if* $T_1 \cap \mathbf{fv}\, P = T_2 \cap \mathbf{fv}\, P$ *and* $\Upsilon_1 \cap \mathbf{fav} P = \Upsilon_2 \cap \mathbf{fav} P$ *then* $T_1 ; \Upsilon_1 \vdash P \iff T_2 ; \Upsilon_2 \vdash P$*, and*
− *if* $T_1 \cap \mathbf{fv}\, N = T_2 \cap \mathbf{fv}\, N$ *and* $\Upsilon_1 \cap \mathbf{fav} N = \Upsilon_2 \cap \mathbf{fav} N$ *then* $T_1 ; \Upsilon_1 \vdash N \iff T_2 ; \Upsilon_2 \vdash N$.

Proof. By mutual structural induction on P and N. □

Lemma 52 (Variable algorithmization agrees with well-formedness).

+ $T, \overrightarrow{\alpha^-} \vdash P$ *implies* $T ; \overrightarrow{\widehat{\alpha}^-} \vdash [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]P$;
− $T, \overrightarrow{\alpha^-} \vdash N$ *implies* $T ; \overrightarrow{\widehat{\alpha}^-} \vdash [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]N$.

Proof. The proof is a structural induction on $T, \overrightarrow{\alpha^-} \vdash P$ and mutually, on $T, \overrightarrow{\alpha^-} \vdash N$. Notice that the substitutions commute with all the constructors, providing the step of the induction. □

Lemma 53 (Variable de-algorithmization agrees with well-formedness).

+ $T ; \overrightarrow{\widehat{\alpha}^-} \vdash P$ *implies* $T, \overrightarrow{\alpha^-} \vdash [\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}]P$;
− $T ; \overrightarrow{\widehat{\alpha}^-} \vdash N$ *implies* $T, \overrightarrow{\alpha^-} \vdash [\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}]N$.

Proof. As for lemma 52, the proof is a structural induction on $T ; \overrightarrow{\widehat{\alpha}^-} \vdash P$ and mutually, on $T ; \overrightarrow{\widehat{\alpha}^-} \vdash N$. □

Corollary 22 (Well-formedness Algorithmic Context Weakening). *Suppose that* $T_1 \subseteq T_2$*, and* $\Upsilon_1 \subseteq \Upsilon_2$*. Then*

+ *if* $T_1 ; \Upsilon_1 \vdash P$ *implies* $T_2 ; \Upsilon_2 \vdash P$,
− *if* $T_1 ; \Upsilon_1 \vdash N$ *implies* $T_2 ; \Upsilon_2 \vdash N$.

Proof. By lemma 50, $T_1 ; \Upsilon_1 \vdash P$ implies $\mathbf{fv}\,(P) \subseteq T_1 \subseteq T_2$ and $\mathbf{fav}(P) \subseteq \Upsilon_1 \subseteq \Upsilon_2$, and thus, $\mathbf{fv}\,(P) = \mathbf{fv}\,(P) \cap T_1 = \mathbf{fv}\,(P) \cap T_2$, and $\mathbf{fav}(P) = \mathbf{fav}(P) \cap \Upsilon_1 = \mathbf{fav}(P) \cap \Upsilon_2$. Then by lemma 51, $T_2 ; \Upsilon_2 \vdash P$. The negative case is symmetric. □

9.2 Substitution

Lemma 55 (Algorithmic Substitution Strengthening). *Restricting the substitution to the algorithmic variables of the substitution subject does not affect the result. Suppose that* $\widehat{\sigma}$ *is an algorithmic substitution,* P *and* N *are algorithmic types. Then*

+ $[\widehat{\sigma}]P = [\widehat{\sigma}|_{\mathbf{fav}\, P}]P$,
− $[\widehat{\sigma}]N = [\widehat{\sigma}|_{\mathbf{fav}\, N}]N$

Proof. The proof is analogous to the proof of lemma 4. □

Lemma 56 (Substitutions equal on the algorithmic variables). *Suppose that* $\widehat{\sigma}_1$ *and* $\widehat{\sigma}_2$ *are normalized substitutions of signature* $\Sigma \vdash \widehat{\sigma}_i : \Upsilon$*. Then*

+ *for a normalized type* $T ; \Upsilon \vdash P$*, if* $[\widehat{\sigma}_1]P = [\widehat{\sigma}_2]P$ *then* $\widehat{\sigma}_1|_{(\mathbf{fav}\, P)} = \widehat{\sigma}_2|_{(\mathbf{fav}\, P)}$;
− *for a normalized type* $T ; \Upsilon \vdash N$*, if* $[\widehat{\sigma}_1]N = [\widehat{\sigma}_2]N$ *then* $\widehat{\sigma}_1|_{(\mathbf{fav}\, N)} = \widehat{\sigma}_2|_{(\mathbf{fav}\, N)}$.

Proof. The proof is a simple structural induction on $T ; \Upsilon \vdash P$ and mutually, on $T ; \Upsilon \vdash N$. Let us consider the shape of N (the cases of P are symmetric).

Case 1. $T ; \Upsilon \vdash \widehat{\alpha}^-$. Then $[\widehat{\sigma}_1]\widehat{\alpha}^- = [\widehat{\sigma}_2]\widehat{\alpha}^-$ implies $\widehat{\sigma}_1|_{(\mathbf{fav}\,\widehat{\alpha}^-)} = \widehat{\sigma}_2|_{(\mathbf{fav}\,\widehat{\alpha}^-)}$ immediately.

Case 2. $T\,;\Upsilon \vdash \widehat{\alpha}^-$. Then $\mathbf{fav}\widehat{\alpha}^- = \emptyset$, and $\widehat{\sigma}_1|_{(\mathbf{fav}\widehat{\alpha}^-)} = \widehat{\sigma}_2|_{(\mathbf{fav}\widehat{\alpha}^-)}$ holds vacuously.

Case 3. $T\,;\Upsilon \vdash \forall\overrightarrow{\alpha^+}.\ N$. Then we are proving that $[\widehat{\sigma}_1]\forall\overrightarrow{\alpha^+}.\ N = [\widehat{\sigma}_2]\forall\overrightarrow{\alpha^+}.\ N$ implies $\widehat{\sigma}_1|_{(\mathbf{fav}\forall\overrightarrow{\alpha^+}.\ N)} = \widehat{\sigma}_2|_{(\mathbf{fav}\forall\overrightarrow{\alpha^+}.\ N)}$. By definition of substitution and ($\forall^{\simeq^D}$), $[\widehat{\sigma}_1]N = [\widehat{\sigma}_2]N$ implies $\widehat{\sigma}_1|_{\mathbf{fav}\,N} = \widehat{\sigma}_2|_{\mathbf{fav}\,N}$. Since $\forall\overrightarrow{\alpha^+}.\ N$ is normalized, so is $T, \overrightarrow{\alpha^+}\,;\Upsilon \vdash N$, hence, the induction hypothesis is applicable and implies $\widehat{\sigma}_1|_{\mathbf{fav}\,N} = \widehat{\sigma}_2|_{\mathbf{fav}\,N}$, as required.

Case 4. $T\,;\Upsilon \vdash P \to N$. Then we are proving that $[\widehat{\sigma}_1](P \to N) = [\widehat{\sigma}_2](P \to N)$ implies $\widehat{\sigma}_1|_{(\mathbf{fav}\,P\to N)} = \widehat{\sigma}_2|_{(\mathbf{fav}\,P\to N)}$. By definition of substitution and congruence of equality, $[\widehat{\sigma}_1](P \to N) = [\widehat{\sigma}_2](P \to N)$ means $[\widehat{\sigma}_1]P = [\widehat{\sigma}_2]P$ and $[\widehat{\sigma}_1]N = [\widehat{\sigma}_2]N$. Notice that P and N are normalized since $P \to N$ is normalized, and well-formed in the same contexts. This way, by the induction hypothesis, $\widehat{\sigma}_1|_{(\mathbf{fav}\,P)} = \widehat{\sigma}_2|_{(\mathbf{fav}\,P)}$ and $\widehat{\sigma}_1|_{(\mathbf{fav}\,N)} = \widehat{\sigma}_2|_{(\mathbf{fav}\,N)}$, which since $\mathbf{fav}(P \to N) = \mathbf{fav}\,P \cup \mathbf{fav}\,N$ implies $\widehat{\sigma}_1|_{(\mathbf{fav}\,P\to N)} = \widehat{\sigma}_2|_{(\mathbf{fav}\,P\to N)}$.

Case 5. $T\,;\Upsilon \vdash \uparrow P$. The proof is similar to the previous case: we apply congruence of substitution, equality, and normalization, then the induction hypothesis, and then the fact that $\mathbf{fav}(\uparrow P) = \mathbf{fav}\,P$.

□

Corollary 23 (Substitutions equivalent on the algorithmic variables). *Suppose that $\widehat{\sigma}_1$ and $\widehat{\sigma}_2$ are substitutions of signature $\Sigma \vdash \widehat{\sigma}_i : \Upsilon$ where $T \vdash^{\supseteq} \Sigma$. Then*

+ *for a type $T\,;\Upsilon \vdash P$, if $T \vdash [\widehat{\sigma}_1]P \simeq^{\leqslant} [\widehat{\sigma}_2]P$ then $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{fav}\,P$;*
− *for a type $T\,;\Upsilon \vdash N$, if $T \vdash [\widehat{\sigma}_1]N \simeq^{\leqslant} [\widehat{\sigma}_2]N$ then $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{fav}\,N$.*

Proof. First, let us normalize the types and the substitutions, and show that the given equivalences and well-formedness properties are preserved. $T\,;\Upsilon \vdash P$ implies $T\,;\Upsilon \vdash \mathbf{nf}\,(P)$ by corollary 24. $\Sigma \vdash\ \ [\widehat{\sigma}_1]P \simeq^D [\widehat{\sigma}_2]P$ implies $\mathbf{nf}\,([\widehat{\sigma}_1]P) = \mathbf{nf}\,([\widehat{\sigma}_2]P)$ by lemma 46. Then $\mathbf{nf}\,([\widehat{\sigma}_1]P) = \mathbf{nf}\,([\widehat{\sigma}_2]P)$ implies $[\mathbf{nf}\,(\widehat{\sigma}_1)]\mathbf{nf}\,(P) = [\mathbf{nf}\,(\widehat{\sigma}_2)]\mathbf{nf}\,(P)$ by lemma 41. Notice that by corollary 25 $\Sigma \vdash \widehat{\sigma}_i : \Upsilon$ implies $\Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}_i) : \Upsilon$.

This way, by lemma 56, $\Sigma \vdash\ \ [\widehat{\sigma}_1]P \simeq^D [\widehat{\sigma}_2]P$ implies $\mathbf{nf}\,(\widehat{\sigma}_1)|_{(\mathbf{favnf}\,(P))} = \mathbf{nf}\,(\widehat{\sigma}_2)|_{(\mathbf{favnf}\,(P))}$. Then by lemma 58, $\mathbf{nf}\,(\widehat{\sigma}_1)|_{(\mathbf{fav}\,P)} = \mathbf{nf}\,(\widehat{\sigma}_2)|_{(\mathbf{fav}\,P)}$, and by corollary 26, $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{fav}\,P$.

Symmetrically, $\Sigma \vdash\ \ [\widehat{\sigma}_1]N \simeq^D [\widehat{\sigma}_2]N$ implies $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{fav}\,N$. □

9.3 Normalization

Lemma 58 (Algorithmic variables are not changed by the normalization).

− $\mathbf{fav}\,N \equiv \mathbf{favnf}\,(N)$
+ $\mathbf{fav}\,P \equiv \mathbf{favnf}\,(P)$

Proof. By straightforward induction on N and mutually on P, similar to the proof of lemma 38.

□

Lemma 59 (Soundness of normalization of algorithmic types).

− $N \simeq^D \mathbf{nf}\,(N)$
+ $P \simeq^D \mathbf{nf}\,(P)$

Proof. The proof coincides with the proof of lemma 39. □

9.4 Equivalence

Lemma 60 (Algorithmic type well-formedness is invariant under equivalence). *Mutual subtyping implies declarative equivalence.*

+ *if $P \simeq^D Q$ then $T\,;\Upsilon \vdash P \iff T\,;\Upsilon \vdash Q$,*

– *if* $N \simeq^D M$ *then* $T\,;\Upsilon \vdash N \iff T\,;\Upsilon \vdash M$

Proof. The proof coincides with the proof of lemma 26, and adds two cases for equating two positive or two negative algorithmic variables, which must be equal by inversion, and thus, $T\,;\Upsilon \vdash \widehat{\alpha}^{\pm} \iff T\,;\Upsilon \vdash \widehat{\alpha}^{\pm}$ holds trivially. □

Corollary 24 (Normalization preserves well-formedness of algorithmic types).

+ $T\,;\Upsilon \vdash P \iff T\,;\Upsilon \vdash \mathbf{nf}\,(P)$,
– $T\,;\Upsilon \vdash N \iff T\,;\Upsilon \vdash \mathbf{nf}\,(N)$

Proof. Immediately from lemmas 59 and 60. □

Corollary 25 (Normalization preserves the signature of the algorithmic substitution).
$\Sigma \vdash \widehat{\sigma} : \Upsilon \iff \Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}) : \Upsilon, T \vdash \widehat{\sigma} : \Upsilon \iff T \vdash \mathbf{nf}\,(\widehat{\sigma}) : \Upsilon$.

Proof. The proof is analogous to corollary 15. □

Corollary 26 (Algorithmic substitution equivalence becomes equality after normalization). *Suppose that* $\Sigma \vdash \widehat{\sigma}_1 : \Upsilon'$ *and* $\Sigma \vdash \widehat{\sigma}_2 : \Upsilon'$ *are algorithmic substitutions and* $\Upsilon \subseteq \Upsilon'$. *Then* $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \Upsilon \iff \mathbf{nf}\,(\widehat{\sigma}_1)|_{\Upsilon} = \mathbf{nf}\,(\widehat{\sigma}_2)|_{\Upsilon}$.

Proof. Follows immediately from lemma 46:

$\Rightarrow$ If $\widehat{\alpha}^{\pm} \notin \Upsilon$, then $[\mathbf{nf}\,(\widehat{\sigma}_1)|_{\Upsilon}]\widehat{\alpha}^{\pm} = [\mathbf{nf}\,(\widehat{\sigma}_2)|_{\Upsilon}]\widehat{\alpha}^{\pm} = \widehat{\alpha}^{\pm}$ by definition. For any $\widehat{\alpha}^{\pm} \in \Upsilon$, $[\mathbf{nf}\,(\widehat{\sigma}_1)|_{\Upsilon}]\widehat{\alpha}^{\pm} = \mathbf{nf}\,([\widehat{\sigma}_1]\widehat{\alpha}^{\pm})$ and $[\mathbf{nf}\,(\widehat{\sigma}_2)|_{\Upsilon}]\widehat{\alpha}^{\pm} = \mathbf{nf}\,([\widehat{\sigma}_2]\widehat{\alpha}^{\pm})$; $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}_1]\widehat{\alpha}^{\pm} \simeq^{\leqslant} [\widehat{\sigma}_2]\widehat{\alpha}^{\pm}$ implies $\mathbf{nf}\,([\widehat{\sigma}_1]\widehat{\alpha}^{\pm}) = \mathbf{nf}\,([\widehat{\sigma}_2]\widehat{\alpha}^{\pm})$ by lemma 43.

$\Leftarrow$ If $\widehat{\alpha}^{\pm} \in \Upsilon$, then $\mathbf{nf}\,(\widehat{\sigma}_1)|_{\Upsilon} = \mathbf{nf}\,(\widehat{\sigma}_2)|_{\Upsilon}$ implies $\mathbf{nf}\,([\widehat{\sigma}_1]\widehat{\alpha}^{\pm}) = \mathbf{nf}\,([\widehat{\sigma}_2]\widehat{\alpha}^{\pm})$ by definition of substitution restriction and normalization. In turn, $\mathbf{nf}\,([\widehat{\sigma}_1]\widehat{\alpha}^{\pm}) = \mathbf{nf}\,([\widehat{\sigma}_2]\widehat{\alpha}^{\pm})$ means $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}_1]\widehat{\alpha}^{\pm} \simeq^{\leqslant} [\widehat{\sigma}_2]\widehat{\alpha}^{\pm}$ by lemma 43.

□

9.5 Unification Constraint Merge

Observation 3 (Unification Constraint Merge Determinism). *Suppose that* $\Sigma \vdash UC_1$ *and* $\Sigma \vdash UC_2$ *If* $\Sigma \vdash UC_1 \mathbin{\&} UC_2 = UC$ *and* $\Sigma \vdash UC_1 \mathbin{\&} UC_2 = UC'$ *are defined then* $UC = UC'$.

Proof. UC and UC' both consists of three parts: Entries of UC_1 that do not have matching entries in UC_2, entries of UC_2 that do not have matching entries in UC_1, and the merge of matching entries.

The parts corresponding to unmatched entries of UC_1 and UC_2 coincide, since UC_1 and UC_2 are fixed. To show that the merge of matching entries coincide, let us take any pair of matching $ue_1 \in UC_1$ and $ue_2 \in UC_2$ and consider their shape.

Case 1. ue_1 is $\widehat{\alpha}^{+} :\simeq Q_1$ and ue_2 is $\widehat{\alpha}^{+} :\simeq Q_2$ then the result, if it exists, is always ue_1, by inversion of $(\simeq \mathbin{\&}^{+} \simeq)$.

Case 2. ue_1 is $\widehat{\alpha}^{-} :\simeq N_1$ and ue_2 is $\widehat{\alpha}^{-} :\simeq N_2$ then analogously, the result, if it exists, is always ue_1, by inversion of $(\simeq \mathbin{\&}^{-} \simeq)$.

This way, the third group of entries coincide as well. □

Lemma 61 (Soundness of Unification Constraint Merge). *Suppose that* $\Sigma \vdash UC_1$ *and* $\Sigma \vdash UC_2$ *are normalized unification constraints. If* $\Sigma \vdash UC_1 \mathbin{\&} UC_2 = UC$ *is defined then* $UC = UC_1 \cup UC_2$.

Proof.

- $UC_1 \mathbin{\&} UC_2 \subseteq UC_1 \cup UC_2$
  By definition, $UC_1 \mathbin{\&} UC_2$ consists of three parts: entries of UC_1 that do not have matching entries of UC_2, entries of UC_2 that do not have matching entries of UC_1, and the merge of matching entries.
  If ue is from the first or the second part, then $ue \in UC_1 \cup UC_2$ holds immediately. If ue is from the third part, then ue is the merge of two matching entries $ue_1 \in UC_1$ and $ue_2 \in UC_2$. Since UC_1 and UC_2 are normalized unification , ue_1 and ue_2 have one of the following forms:
  - $\widehat{\alpha}^+ :\simeq P_1$ and $\widehat{\alpha}^+ :\simeq P_2$, where P_1 and P_2 are normalized, and then since $\Sigma(\widehat{\alpha}^+) \vdash ue_1 \& ue_2 = ue$ exists, $(\simeq \&^+ \simeq)$ was applied to infer it. It means that $ue = ue_1 = ue_2$;
  - $\widehat{\alpha}^- :\simeq N_1$ and $\widehat{\alpha}^- :\simeq N_2$, then symmetrically, $\Sigma(\widehat{\alpha}^-) \vdash ue_1 \mathbin{\&} ue_2 = ue = ue_1 = ue_2$

  In both cases, $ue \in UC_1 \cup UC_2$.
- $UC_1 \cup UC_2 \subseteq UC_1 \mathbin{\&} UC_2$
  Let us take an arbitrary $ue_1 \in UC_1$. Then since UC_1 is a unification constraint, ue_1 has one of the following forms:
  - $\widehat{\alpha}^+ :\simeq P$ where P is normalized. If $\widehat{\alpha}^+ \notin \mathbf{dom}\,(UC_2)$, then $ue_1 \in UC_1 \mathbin{\&} UC_2$. Otherwise, there is a normalized matching $ue_2 = (\widehat{\alpha}^+ :\simeq P') \in UC_2$ and then since $UC_1 \mathbin{\&} UC_2$ exists, $(\simeq \&^+ \simeq)$ was applied to construct $ue_1 \mathbin{\&} ue_2 \in UC_1 \mathbin{\&} UC_2$. By inversion of $(\simeq \&^+ \simeq)$, $ue_1 \mathbin{\&} ue_2 = ue_1$, and $\mathbf{nf}\,(P) = \mathbf{nf}\,(P')$, which since P and P' are normalized, implies that $P = P'$, that is $ue_1 = ue_2 \in UC_1 \mathbin{\&} UC_2$.
  - $\widehat{\alpha}^- :\simeq N$ where N is normalized. Then symmetrically, $ue_1 = ue_2 \in UC_1 \mathbin{\&} UC_2$.

  Similarly, if we take an arbitrary $ue_2 \in UC_2$, then $ue_1 = ue_2 \in UC_1 \mathbin{\&} UC_2$.

□

Corollary 27. *Suppose that $\Sigma \vdash UC_1$ and $\Sigma \vdash UC_2$ are normalized unification constraints. If $\Sigma \vdash UC_1 \mathbin{\&} UC_2 = UC$ is defined then*

(1) $\Sigma \vdash UC$ is normalized unification constraint,
(2) for any substitution $\Sigma \vdash \widehat{\sigma} : \mathbf{dom}\,(UC)$, $\Sigma \vdash \widehat{\sigma} : UC$ implies $\Sigma \vdash \widehat{\sigma} : UC_1$ and $\Sigma \vdash \widehat{\sigma} : UC_2$.

Proof. It is clear that since $UC = UC_1 \cup UC_2$ (by lemma 61), and being normalized means that all entries are normalized, UC is a normalized unification constraint. Analogously, $\Sigma \vdash UC = UC_1 \cup UC_2$ holds immediately, since $\Sigma \vdash UC_1$ and $\Sigma \vdash UC_2$.

Let us take an arbitrary substitution $\Sigma \vdash \widehat{\sigma} : \mathbf{dom}\,(UC)$ and assume that $\Sigma \vdash \widehat{\sigma} : UC$. Then $\Sigma \vdash \widehat{\sigma} : UC_i$ holds by definition: If $ue \in UC_i \subseteq UC_1 \cup UC_2 = UC$ then $\Sigma(\widehat{\alpha}^\pm) \vdash [\widehat{\sigma}]\widehat{\alpha}^\pm : ue$ (where ue restricts $\widehat{\alpha}^\pm$) holds since $\Sigma \vdash \widehat{\sigma} : \mathbf{dom}\,(UC)$. □

Lemma 62 (Completeness of Unification Constraint Entry Merge). *For a fixed context T, suppose that $T \vdash ue_1$ and $T \vdash ue_2$ are matching constraint entries.*

+ *for a type P such that $T \vdash P : ue_1$ and $T \vdash P : ue_2$, $T \vdash ue_1 \& ue_2 = ue$ is defined and $T \vdash P : ue$.*
− *for a type N such that $T \vdash N : ue_1$ and $T \vdash N : ue_2$, $T \vdash ue_1 \mathbin{\&} ue_2 = ue$ is defined and $T \vdash N : ue$.*

Proof. Let us consider the shape of ue_1 and ue_2.

Case 1. ue_1 is $\widehat{\alpha}^+ :\simeq Q_1$ and ue_2 is $\widehat{\alpha}^+ :\simeq Q_2$. Then $T \vdash P : ue_1$ means $T \vdash P \simeq^\leqslant Q_1$, and $T \vdash P : ue_2$ means $T \vdash P \simeq^\leqslant Q_2$. Then by transitivity of equivalence (corollary 10), $T \vdash Q_1 \simeq^\leqslant Q_2$, which means $\mathbf{nf}\,(Q_1) = \mathbf{nf}\,(Q_2)$ by lemma 46. Hence, $(\simeq \&^+ \simeq)$ applies to infer $T \vdash ue_1 \& ue_2 = ue_2$, and $T \vdash P : ue_2$ holds by assumption.

Case 2. ue_1 is $\widehat{\alpha}^- :\simeq N_1$ and ue_2 is $\widehat{\alpha}^- :\simeq M_2$. The proof is symmetric.

□

Lemma 63 (Completeness of Unification Constraint Merge). *Suppose that $\Sigma \vdash UC_1$ and $\Sigma \vdash UC_2$. Then for any $\Upsilon \supseteq \mathbf{dom}\,(UC_1) \cup \mathbf{dom}\,(UC_2)$ and substitution $\Sigma \vdash \widehat{\sigma} : \Upsilon$ such that $\Sigma \vdash \widehat{\sigma} : UC_1$ and $\Sigma \vdash \widehat{\sigma} : UC_2$,*

(1) $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$ is defined and
(2) $\Sigma \vdash \widehat{\sigma} : UC$.

Proof. The proof repeats the proof of lemma 83 for cases uses lemma 62 instead of lemma 82. □

9.6 Unification

Observation 4 (Unification Determinism).

+ *If $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$ and $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC'$ then $UC = UC'$.*
− *If $T;\Sigma \vDash N \overset{u}{\simeq} M \dashv UC$ and $T;\Sigma \vDash N \overset{u}{\simeq} M \dashv UC'$ then $UC = UC'$.*

Proof. We prove it by mutual structural induction on $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$ and $T;\Sigma \vDash N \overset{u}{\simeq} M \dashv UC'$. Let us consider the positive case only since the negative case is symmetric.

First, notice that the rule applied the last uniquely determined by the shape of P and Q. Second, the premises of each rule are deterministic on the input either by the induction hypothesis or by observation 3. □

Lemma 64 (Soundness of Unification).

+ *For normalized P and Q such that $T\,;\mathbf{dom}\,(\Sigma) \vdash P$ and $T \vdash Q$,*
  if $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$ then $\Sigma \vdash UC : \mathbf{fav}\,P$ and for any normalized $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : UC$, $[\widehat{\sigma}]P = Q$.
− *For normalized N and M such that $T\,;\mathbf{dom}\,(\Sigma) \vdash N$ and $T \vdash M$,*
  if $T;\Sigma \vDash N \overset{u}{\simeq} M \dashv UC$ then $\Sigma \vdash UC : \mathbf{fav}\,N$ and for any normalized $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : UC$, $[\widehat{\sigma}]N = M$.

Proof. We prove by induction on the derivation of $T;\Sigma \vDash N \overset{u}{\simeq} M \dashv UC$ and mutually $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$. Let us consider the last rule forming this derivation.

Case 1. ($\text{Var}^{\overset{u}{\simeq}}_{-}$), then $N = \alpha^- = M$. The resulting unification constraint is empty: $UC = \cdot$. It satisfies $\Sigma \vdash UC : \cdot$ vacuously, and $[\widehat{\sigma}]\alpha^- = \alpha^-$, that is $[\widehat{\sigma}]N = M$.

Case 2. ($\uparrow^{\overset{u}{\simeq}}$), then $N = \uparrow P$ and $M = \uparrow Q$. The algorithm makes a recursive call to $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC$ returning UC. By induction hypothesis, $\Sigma \vdash UC : \mathbf{fav}\,P$ and thus, $\Sigma \vdash UC : \mathbf{fav}\uparrow P$, and for any $\Sigma \vdash \widehat{\sigma} : UC$, $[\widehat{\sigma}]N = [\widehat{\sigma}]\uparrow P = \uparrow[\widehat{\sigma}]P = \uparrow Q = M$, as required.

Case 3. ($\to^{\overset{u}{\simeq}}$), then $N = P \to N'$ and $M = Q \to M'$. The algorithm makes two recursive calls to $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC_1$ and $T;\Sigma \vDash N' \overset{u}{\simeq} M' \dashv UC_2$ returning $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$ as the result.

It is clear that P, N', Q, and M' are normalized, and that $T\,;\mathbf{dom}\,(\Sigma) \vdash P$, $T\,;\mathbf{dom}\,(\Sigma) \vdash N'$, $T \vdash Q$, and $T \vdash M'$. This way, the induction hypothesis is applicable to both recursive calls.

By applying the induction hypothesis to $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC_1$, we have:

- $\Sigma \vdash UC_1 : \mathbf{fav}\,P$,
- for any $\Sigma \vdash \widehat{\sigma}' : UC_1$, $[\widehat{\sigma}']P = Q$.

By applying it to $T;\Sigma \vDash N' \overset{u}{\simeq} M' \dashv UC_2$, we have:

- $\Sigma \vdash UC_2 : \mathbf{fav}\,N'$,
- for any $\Sigma \vdash \widehat{\sigma}' : UC_2$, $[\widehat{\sigma}']N' = M'$.

Let us take an arbitrary $\Sigma \vdash \widehat{\sigma} : UC$. By the soundness of the constraint merge (lemma 81), $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$ implies $\Sigma \vdash \widehat{\sigma} : UC_1$ and $\Sigma \vdash \widehat{\sigma} : UC_2$.

Applying the induction hypothesis to $\Sigma \vdash \widehat{\sigma} : UC_1$, we have $[\widehat{\sigma}]P = Q$; applying it to $\Sigma \vdash \widehat{\sigma} : UC_2$, we have $[\widehat{\sigma}]N' = M'$. This way, $[\widehat{\sigma}]N = [\widehat{\sigma}]P \to [\widehat{\sigma}]N' = Q \to M' = M$.

Case 4. ($\forall^{\overset{u}{\simeq}}$), then $N = \forall\overrightarrow{\alpha^+}. N'$ and $M = \forall\overrightarrow{\alpha^+}. M'$. The algorithm makes a recursive call to $T, \overrightarrow{\alpha^+}; \Sigma \models N' \overset{u}{\simeq} M' \dashv UC$ returning UC as the result.

The induction hypothesis is applicable: $T, \overrightarrow{\alpha^+}\,;\, \mathbf{dom}\,(\Sigma) \vdash N'$ and $T, \overrightarrow{\alpha^+} \vdash M'$ hold by inversion, and N' and M' are normalized, since N and M are. Let us take an arbitrary $\Sigma \vdash \widehat{\sigma} : UC$. By the induction hypothesis, $[\widehat{\sigma}]N' = M'$. Then $[\widehat{\sigma}]N = [\widehat{\sigma}]\forall\overrightarrow{\alpha^+}. N' = \forall\overrightarrow{\alpha^+}. [\widehat{\sigma}]N' = \forall\overrightarrow{\alpha^+}. M' = M$.

Case 5. ($\text{UVar}^{\overset{u}{\simeq}}$), then $N = \widehat{\alpha}^-$, $\widehat{\alpha}^-\{\Theta\} \in \Sigma$, and $\Theta \vdash M$. As the result, the algorithm returns $UC = (\widehat{\alpha}^- :\simeq M)$.

It is clear that $\widehat{\alpha}^-\{\Theta\} \vdash (\widehat{\alpha}^- :\simeq M)$, since $\Theta \vdash M$, meaning that $\Sigma \vdash UC$.

Let us take an arbitrary $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : UC$. Since $UC = (\widehat{\alpha}^- :\simeq M)$, $\Sigma \vdash \widehat{\sigma} : UC$ implies $\Sigma(\widehat{\alpha}^-) \vdash [\widehat{\sigma}]\widehat{\alpha}^- : (\widehat{\alpha}^- :\simeq M)$. By inversion of ($:\simeq^{\text{SAT}}$), it means $\Sigma(\widehat{\alpha}^-) \vdash [\widehat{\sigma}]\widehat{\alpha}^- \simeq^{\leqslant} M$. This way, $\Sigma(\widehat{\alpha}^-) \vdash [\widehat{\sigma}]N \simeq^{\leqslant} M$. Notice that $\widehat{\sigma}$ and N are normalized, and by lemma 41, so is $[\widehat{\sigma}]N$. Since both sides of $\Sigma(\widehat{\alpha}^-) \vdash [\widehat{\sigma}]N \simeq^{\leqslant} M$ are normalized, by lemma 46, we have $[\widehat{\sigma}]N = M$.

Case 6. The positive cases are proved symmetrically.

□

Lemma 65 (Completeness of Unification).

+ *For normalized P and Q such that $T\,;\,$**dom**$\,(\Sigma) \vdash P$ and $T \vdash Q$, suppose that there exists $\Sigma \vdash \widehat{\sigma} :$ **fav**(P) such that $[\widehat{\sigma}]P = Q$, then $T; \Sigma \models P \overset{u}{\simeq} Q \dashv UC$ for some UC.*
− *For normalized N and M such that $T\,;\,$**dom**$\,(\Sigma) \vdash N$ and $T \vdash M$, suppose that there exists $\Sigma \vdash \widehat{\sigma} :$ **fav**(N) such that $[\widehat{\sigma}]N = M$, then $T; \Sigma \models N \overset{u}{\simeq} M \dashv UC$ for some UC.*

Proof. We prove it by induction on the structure of P and mutually, N.

Case 1. $N = \widehat{\alpha}^-$

$T\,;\,\mathbf{dom}\,(\Sigma) \vdash \widehat{\alpha}^-$ means that $\widehat{\alpha}^-\{\Theta\} \in \Sigma$ for some Θ.

Let us take an arbitrary $\Sigma \vdash \widehat{\sigma} : \widehat{\alpha}^-$ such that $[\widehat{\sigma}]\widehat{\alpha}^- = M$. $\Sigma \vdash \widehat{\sigma} : \widehat{\alpha}^-$ means that $\Theta \vdash M$.

This way, ($\text{UVar}^{\overset{u}{\simeq}}$) is applicable to infer $T; \Sigma \models \widehat{\alpha}^- \overset{u}{\simeq} M \dashv (\widehat{\alpha}^- :\simeq M)$.

Case 2. $N = \alpha^-$

Let us take an arbitrary $\Sigma \vdash \widehat{\sigma} :$ **fav**(α^-) such that $[\widehat{\sigma}]\alpha^- = M$. Since **fav**$(\alpha^-) = \emptyset$, $[\widehat{\sigma}]\alpha^- = M$ means $M = \alpha^-$.

This way, ($\text{Var}^{\overset{u}{\simeq}}$) infers $T; \Sigma \models \alpha^- \overset{u}{\simeq} \alpha^- \dashv \cdot$, which is rewritten as $T; \Sigma \models N \overset{u}{\simeq} M \dashv \cdot$.

Case 3. $N = \uparrow P$

Let us take an arbitrary $\Sigma \vdash \widehat{\sigma} :$ **fav**(P) such that $[\widehat{\sigma}]\uparrow P = M$. The latter means $\uparrow[\widehat{\sigma}]P = M$, i.e. $M = \uparrow Q$ for some Q and $[\widehat{\sigma}]P = Q$.

Let us show that the induction hypothesis is applicable to $[\widehat{\sigma}]P = Q$. Notice that P is normalized, since $N = \uparrow P$ is normalized, $T\,;\mathbf{dom}\,(\Sigma) \vdash P$ holds by inversion of $T\,;\mathbf{dom}\,(\Sigma) \vdash \uparrow P$, and $T \vdash Q$ holds by inversion of $T \vdash \uparrow Q$.

This way, by the induction hypothesis there exists UC such that $T; \Sigma \models P \overset{u}{\simeq} Q \dashv UC$.

Case 4. $N = P \to N'$

Let us take an arbitrary $\Sigma \vdash \widehat{\sigma} :$ **fav**$(P \to N')$ such that $[\widehat{\sigma}](P \to N') = M$. The latter means $[\widehat{\sigma}]P \to [\widehat{\sigma}]N' = M$, i.e. $M = Q \to M'$ for some Q and M', such that $[\widehat{\sigma}]P = Q$ and $[\widehat{\sigma}]N' = M'$.

Let us show that the induction hypothesis is applicable to $\Sigma \vdash \widehat{\sigma}|_{\mathbf{fav}(P)} : \mathbf{fav}(P)$ and $[\widehat{\sigma}|_{\mathbf{fav}(P)}]P = Q$ (the latter holds since $[\widehat{\sigma}|_{\mathbf{fav}(P)}]P = [\widehat{\sigma}]P$ by lemma 55),

- P is normalized, since $N = P \to N'$ is normalized
- $T\,;\mathbf{dom}\,(\Sigma) \vdash P$ follows from the inversion of $T\,;\mathbf{dom}\,(\Sigma) \vdash P \to N'$,
- $T \vdash Q$.

Then by the induction hypothesis, $T;\Sigma \vDash P \overset{u}{\simeq} Q \dashv UC_1$. Analogously, the induction hypothesis is applicable to $[\widehat{\sigma}|_{\mathbf{fav}N'}]N' = M'$, and thus, $T;\Sigma \vDash N' \overset{u}{\simeq} M' \dashv UC_2$.

To apply $(\to^{\overset{u}{\simeq}})$ and infer the required $T;\Sigma \vDash N \overset{u}{\simeq} M \dashv UC$, it is left to show that $\Sigma \vdash UC_1 \,\&\, UC_2 = UC$. It holds by completeness of the unification constraint merge (lemma 63) for $\Sigma \vdash UC_1 : \mathbf{fav}\,P$, $\Sigma \vdash UC_2 : \mathbf{fav}\,N'$ (which hold by soundness), and $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P) \cup \mathbf{fav}(N')$, which holds since $\mathbf{fav}(P) \cup \mathbf{fav}(N') = \mathbf{fav}(P \to N')$. Notice that by soundness, $\Sigma \vdash \widehat{\sigma}|_{\mathbf{fav}(P)} : UC_1$, which implies $\Sigma \vdash \widehat{\sigma} : UC_1$. Analogously, $\Sigma \vdash \widehat{\sigma} : UC_2$.

Case 5. $N = \forall\overrightarrow{\alpha^+}.\, N'$

Let us take an arbitrary $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N')$ such that $[\widehat{\sigma}]\forall\overrightarrow{\alpha^+}.\, N' = M$. The latter means $\forall\overrightarrow{\alpha^+}.\, [\widehat{\sigma}]N' = M$, i.e. $M = \forall\overrightarrow{\alpha^+}.\, M'$ for some M' such that $[\widehat{\sigma}]N' = M'$.

Let us show that the induction hypothesis is applicable to $[\widehat{\sigma}]N' = M'$. Notice that N' is normalized, since $N = \forall\overrightarrow{\alpha^+}.\, N'$ is normalized, $T, \overrightarrow{\alpha^+}\,;\mathbf{dom}\,(\Sigma) \vdash N'$ follows from inversion of $T\,;\mathbf{dom}\,(\Sigma) \vdash \forall\overrightarrow{\alpha^+}.\, N'$, $T, \overrightarrow{\alpha^+} \vdash M'$ follows from inversion of $T \vdash \forall\overrightarrow{\alpha^+}.\, M'$, and $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N')$ by assumption.

This way, by the induction hypothesis, $T, \overrightarrow{\alpha^+};\Sigma \vDash N' \overset{u}{\simeq} M' \dashv UC$ exists and moreover, $\Sigma \vdash \widehat{\sigma} : UC$. Hence, $(\forall^{\overset{u}{\simeq}})$ is applicable to infer $T;\Sigma \vDash \forall\overrightarrow{\alpha^+}.\, N' \overset{u}{\simeq} \forall\overrightarrow{\alpha^+}.\, M' \dashv UC$, that is $T;\Sigma \vDash N \overset{u}{\simeq} M \dashv UC$.

Case 6. The positive cases are proved symmetrically.

□

9.7 Anti-unification

Observation 5 (Determinism of Anti-unification Algorithm).

+ *If* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ *and* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$, *then* $\Upsilon = \Upsilon'$, $Q = Q'$, $\widehat{\tau}_1 = \widehat{\tau}'_1$, *and* $\widehat{\tau}_2 = \widehat{\tau}'_2$.
− *If* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ *and* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon', M', \widehat{\tau}'_1, \widehat{\tau}'_2)$, *then* $\Upsilon = \Upsilon'$, $M = M'$, $\widehat{\tau}_1 = \widehat{\tau}'_1$, *and* $\widehat{\tau}_2 = \widehat{\tau}'_2$.

Proof. By trivial induction on $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ and mutually on $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$. □

Observation 6 (Uniqueness of Anti-unification Variable Names). *Names of the anti-unification variables are uniquely defined by the types they are mapped to by the resulting substitutions.*

+ *Assuming* P_1 *and* P_2 *are normalized, if* $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ *then for any* $\widehat{\beta}^- \in \Upsilon$, $\widehat{\beta}^- = \widehat{\alpha}^-_{\{[\widehat{\tau}_1]\widehat{\beta}^-, [\widehat{\tau}_2]\widehat{\beta}^-\}}$
− *Assuming* N_1 *and* N_2 *are normalized, if* $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ *then for any* $\widehat{\beta}^- \in \Upsilon$, $\widehat{\beta}^- = \widehat{\alpha}^-_{\{[\widehat{\tau}_1]\widehat{\beta}^-, [\widehat{\tau}_2]\widehat{\beta}^-\}}$

Proof. By simple induction on $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ and mutually on $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$. Let us consider the last rule applied to infer this judgment.

Case 1. ($\text{Var}^{\overset{a}{\simeq}}_{+}$) or ($\text{Var}^{\overset{a}{\simeq}}_{-}$), then $\Upsilon = \cdot$, and the property holds vacuously.

Case 2. (AU) Then $\Upsilon = \widehat{\alpha}^{-}_{\{N_1,N_2\}}$, $\widehat{\tau}_1 = \widehat{\alpha}^{-}_{\{N_1,N_2\}} \mapsto N_1$, and $\widehat{\tau}_2 = \widehat{\alpha}^{-}_{\{N_1,N_2\}} \mapsto N_2$. So the property holds trivially.

Case 3. ($\to^{\overset{a}{\simeq}}$) In this case, $\Upsilon = \Upsilon' \cup \Upsilon''$, $\widehat{\tau}_1 = \widehat{\tau}'_1 \cup \widehat{\tau}''_1$, and $\widehat{\tau}_2 = \widehat{\tau}'_2 \cup \widehat{\tau}''_2$, where the property holds for $(\Upsilon', \widehat{\tau}'_1, \widehat{\tau}'_2)$ and $(\Upsilon'', \widehat{\tau}''_1, \widehat{\tau}''_2)$ by the induction hypothesis. Then since the union of solutions does not change the types the variables are mapped to, the required property holds for Υ, $\widehat{\tau}_1$, and $\widehat{\tau}_2$.

Case 4. For the other rules, the resulting Υ is taken from the recursive call and the required property holds immediately by the induction hypothesis.

□

Lemma 66 (Soundness of Anti-Unification).

+ *Assuming P_1 and P_2 are normalized, if $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ then*
  (1) $T\,; \Upsilon \vdash Q$,
  (2) $T\,; \cdot \vdash \widehat{\tau}_i : \Upsilon$ for $i \in \{1, 2\}$ are anti-unification substitutions, and
  (3) $[\widehat{\tau}_i]\,Q = P_i$ for $i \in \{1, 2\}$.
– *Assuming N_1 and N_2 are normalized, if $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ then*
  (1) $T\,; \Upsilon \vdash M$,
  (2) $T\,; \cdot \vdash \widehat{\tau}_i : \Upsilon$ for $i \in \{1, 2\}$ are anti-unification substitutions, and
  (3) $[\widehat{\tau}_i]\,M = N_i$ for $i \in \{1, 2\}$.

Proof. We prove it by induction on $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ and mutually, $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$. Let us consider the last rule applied to infer this judgement.

Case 1. ($\text{Var}^{\overset{a}{\simeq}}_{-}$), then $N_1 = \alpha^- = N_2$, $\Upsilon = \cdot$, $M = \alpha^-$, and $\widehat{\tau}_1 = \widehat{\tau}_2 = \cdot$.
(1) $T\,; \cdot \vdash \alpha^-$ follows from the assumption $T \vdash \alpha^-$,
(2) $T\,; \cdot \vdash \cdot : \cdot$ holds trivially, and
(3) $[\cdot]\alpha^- = \alpha^-$ holds trivially.

Case 2. ($\uparrow^{\overset{a}{\simeq}}$), then $N_1 = \uparrow P_1$, $N_2 = \uparrow P_2$, and the algorithm makes the recursive call: $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$, returning $(\Upsilon, \uparrow Q, \widehat{\tau}_1, \widehat{\tau}_2)$ as the result.
Since $N_1 = \uparrow P_1$ and $N_2 = \uparrow P_2$ are normalized, so are P_1 and P_2, and thus, the induction hypothesis is applicable to $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$:
(1) $T\,; \Upsilon \vdash Q$, and hence, $T\,; \Upsilon \vdash \uparrow Q$,
(2) $T\,; \cdot \vdash \widehat{\tau}_i : \Upsilon$ for $i \in \{1, 2\}$, and
(3) $[\widehat{\tau}_i]\,Q = P_i$ for $i \in \{1, 2\}$, and then by the definition of the substitution, $[\widehat{\tau}_i]\uparrow Q = \uparrow P_i$ for $i \in \{1, 2\}$.

Case 3. ($\to^{\overset{a}{\simeq}}$), then $N_1 = P_1 \to N'_1$, $N_2 = P_2 \to N'_2$, and the algorithm makes two recursive calls: $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ and $T \vDash N'_1 \overset{a}{\simeq} N'_2 \dashv (\Upsilon', M, \widehat{\tau}'_1, \widehat{\tau}'_2)$ and and returns $(\Upsilon \cup \Upsilon', Q \to M, \widehat{\tau}_1 \cup \widehat{\tau}'_1, \widehat{\tau}_2 \cup \widehat{\tau}'_2)$ as the result.
Notice that the induction hypothesis is applicable to $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$: P_1 and P_2 are normalized, since $N_1 = P_1 \to N'_1$ and $N_2 = P_2 \to N'_2$ are normalized. Similarly, the induction hypothesis is applicable to $T \vDash N'_1 \overset{a}{\simeq} N'_2 \dashv (\Upsilon', M, \widehat{\tau}'_1, \widehat{\tau}'_2)$.
This way, by the induction hypothesis:

(1) $T\,;\Upsilon \vdash Q$ and $T\,;\Upsilon' \vdash M$. Then by weakening (corollary 22), $T\,;\Upsilon \cup \Upsilon' \vdash Q$ and $T\,;\Upsilon \cup \Upsilon' \vdash M$, which implies $T\,;\Upsilon \cup \Upsilon' \vdash Q \to M$;

(2) $T\,;\cdot \vdash \widehat{\tau}_i : \Upsilon$ and $T\,;\cdot \vdash \widehat{\tau}'_i : \Upsilon'$ Then $T\,;\cdot \vdash \widehat{\tau}_i \cup \widehat{\tau}'_i : \Upsilon \cup \Upsilon'$ are well-defined anti-unification substitutions. Let us take an arbitrary $\widehat{\beta}^- \in \Upsilon \cup \Upsilon'$. If $\widehat{\beta}^- \in \Upsilon$. then $T\,;\cdot \vdash \widehat{\tau}_i : \Upsilon$ implies that $\widehat{\tau}_i$, and hence, $\widehat{\tau}_i \cup \widehat{\tau}'_i$ contains an entry well-formed in T. If $\widehat{\beta}^- \in \Upsilon'$, the reasoning is symmetric.
$\widehat{\tau}_i \cup \widehat{\tau}'_i$ is a well-defined anti-unification substitution: any anti-unification variable occurs uniquely $\widehat{\tau}_i \cup \widehat{\tau}'_i$, since by observation 6, the name of the variable is in one-to-one correspondence with the pair of types it is mapped to by $\widehat{\tau}_1$ and $\widehat{\tau}_2$, an is in one-to-one correspondence with the pair of types it is mapped to by $\widehat{\tau}'_1$ and $\widehat{\tau}'_2$ i.e. if $\widehat{\beta}^- \in \Upsilon \cap \Upsilon'$ then $[\widehat{\tau}_1]\widehat{\beta}^- = [\widehat{\tau}'_1]\widehat{\beta}^-$, and $[\widehat{\tau}_2]\widehat{\beta}^- = [\widehat{\tau}'_2]\widehat{\beta}^-$.

(3) $[\widehat{\tau}_i]\,Q = P_i$ and $[\widehat{\tau}'_i]\,M = N'_i$. Since $\widehat{\tau}_i \cup \widehat{\tau}'_i$ restricted to Υ is $\widehat{\tau}_i$, and $\widehat{\tau}_i \cup \widehat{\tau}'_i$ restricted to Υ' is $\widehat{\tau}'_i$, we have $[\widehat{\tau}_i \cup \widehat{\tau}'_i]\,Q = P_i$ and $[\widehat{\tau}_i \cup \widehat{\tau}'_i]\,M = N'_i$, and thus, $[\widehat{\tau}_i \cup \widehat{\tau}'_i]\,Q \to M = P_1 \to N'_1$

Case 4. $(\forall^{\overset{a}{\simeq}})$, then $N_1 = \forall\overrightarrow{\alpha^+}.\,N'_1$, $N_2 = \forall\overrightarrow{\alpha^+}.\,N'_2$, and the algorithm makes a recursive call: $T \vDash N'_1 \overset{a}{\simeq} N'_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ and returns $(\Upsilon, \forall\overrightarrow{\alpha^+}.\,M, \widehat{\tau}_1, \widehat{\tau}_2)$ as the result.
Similarly to case 2, we apply the induction hypothesis to $T \vDash N'_1 \overset{a}{\simeq} N'_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ to obtain:

(1) $T\,;\Upsilon \vdash M$, and hence, $T\,;\Upsilon \vdash \forall\overrightarrow{\alpha^+}.\,M$;

(2) $T\,;\cdot \vdash \widehat{\tau}_i : \Upsilon$ for $i \in \{1,2\}$, and

(3) $[\widehat{\tau}_i]\,M = N'_i$ for $i \in \{1,2\}$, and then by the definition of the substitution, $[\widehat{\tau}_i]\forall\overrightarrow{\alpha^+}.\,M = \forall\overrightarrow{\alpha^+}.\,N'_i$ for $i \in \{1,2\}$.

Case 5. (AU), which applies when other rules do not, and $T \vdash N_i$, returning as the result $(\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2) = (\widehat{\alpha}^-_{\{N_1,N_2\}}, \widehat{\alpha}^-_{\{N_1,N_2\}}, (\widehat{\alpha}^-_{\{N_1,N_2\}} \mapsto N_1), (\widehat{\alpha}^-_{\{N_1,N_2\}} \mapsto N_2))$.

(1) $T\,;\Upsilon \vdash M$ is rewritten as $T\,;\widehat{\alpha}^-_{\{N_1,N_2\}} \vdash \widehat{\alpha}^-_{\{N_1,N_2\}}$, which holds trivially;

(2) $T\,;\cdot \vdash \widehat{\tau}_i : \Upsilon$ is rewritten as $T\,;\cdot \vdash (\widehat{\alpha}^-_{\{N_1,N_2\}} \mapsto N_i) : \widehat{\alpha}^-_{\{N_1,N_2\}}$, which holds since $T \vdash N_i$ by the premise of the rule;

(3) $[\widehat{\tau}_i]\,M = N_i$ is rewritten as $[\widehat{\alpha}^-_{\{N_1,N_2\}} \mapsto N_i]\widehat{\alpha}^-_{\{N_1,N_2\}} = N_i$, which holds trivially by the definition of substitution.

Case 6. Positive cases are proved symmetrically.

□

Lemma 67 (Completeness of Anti-Unification)**.**

+ *Assume that P_1 and P_2 are normalized, and there exists $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$ such that*

(1) $T\,;\Upsilon' \vdash Q'$,

(2) $T\,;\cdot \vdash \widehat{\tau}'_i : \Upsilon'$ for $i \in \{1,2\}$ are anti-unification substitutions, and

(3) $[\widehat{\tau}'_i]\,Q' = P_i$ for $i \in \{1,2\}$.

Then the anti-unification algorithm terminates, that is there exists $(\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ such that $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$

– *Assume that N_1 and N_2 are normalized, and there exists $(\Upsilon', M', \widehat{\tau}'_1, \widehat{\tau}'_2)$ such that*

(1) $T\,;\Upsilon' \vdash M'$,

(2) $T\,;\cdot \vdash \widehat{\tau}'_i : \Upsilon'$ for $i \in \{1,2\}$, are anti-unification substitutions, and

(3) $[\widehat{\tau}'_i]\,M' = N_i$ for $i \in \{1,2\}$.

Then the anti-unification algorithm succeeds, that is there exists $(\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ such that $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$.

Proof. We prove it by the induction on M' and mutually on Q'.

Case 1. $M' = \widehat{\alpha}^-$ Then since $T\,;\cdot \vdash \widehat{\tau}'_i : \Upsilon'$, $T \vdash [\widehat{\tau}'_i]M' = N_i$. This way, (AU) is always applicable if other rules are not.

Case 2. $M' = \alpha^-$ Then $\alpha^- = [\widehat{\tau}'_i]\alpha^- = N_i$, which means that ($\text{Var}^{\overset{a}{\simeq}}_-$) is applicable.

Case 3. $M' = \uparrow Q'$ Then $\uparrow[\widehat{\tau}'_i]Q' = [\widehat{\tau}'_i]\uparrow Q' = N_i$, that is N_1 and N_2 have form $\uparrow P_1$ and $\uparrow P_2$ respectively.
Moreover, $[\widehat{\tau}'_i]Q' = P_i$, which means that $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$ is an anti-unifier of P_1 and P_2. Then by the induction hypothesis, there exists $(\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ such that $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$, and hence, $T \vDash \uparrow P_1 \overset{a}{\simeq} \uparrow P_2 \dashv (\Upsilon, \uparrow Q, \widehat{\tau}_1, \widehat{\tau}_2)$ by ($\uparrow^{\overset{a}{\simeq}}$).

Case 4. $M' = \forall\overrightarrow{\alpha^+}.\ M''$ This case is similar to the previous one: we consider $\forall\overrightarrow{\alpha^+}$ as a constructor. Notice that $\forall\overrightarrow{\alpha^+}.\ [\widehat{\tau}'_i]M'' = [\widehat{\tau}'_i]\forall\overrightarrow{\alpha^+}.\ M'' = N_i$, that is N_1 and N_2 have form $\forall\overrightarrow{\alpha^+}.\ N''_1$ and $\forall\overrightarrow{\alpha^+}.\ N''_2$ respectively.
Moreover, $[\widehat{\tau}'_i]M'' = N''_i$, which means that $(\Upsilon', M'', \widehat{\tau}'_1, \widehat{\tau}'_2)$ is an anti-unifier of N''_1 and N''_2. Then by the induction hypothesis, there exists $(\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ such that $T \vDash N''_1 \overset{a}{\simeq} N''_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$, and hence, $T \vDash \forall\overrightarrow{\alpha^+}.\ N''_1 \overset{a}{\simeq} \forall\overrightarrow{\alpha^+}.\ N''_2 \dashv (\Upsilon, \forall\overrightarrow{\alpha^+}.\ M, \widehat{\tau}_1, \widehat{\tau}_2)$ by ($\forall^{\overset{a}{\simeq}}$).

Case 5. $M' = Q' \to M''$ Then $[\widehat{\tau}'_i]Q' \to [\widehat{\tau}'_i]M'' = [\widehat{\tau}'_i](Q' \to M'') = N_i$, that is N_1 and N_2 have form $P_1 \to N'_1$ and $P_2 \to N'_2$ respectively.
Moreover, $[\widehat{\tau}'_i]Q' = P_i$ and $[\widehat{\tau}'_i]M'' = N''_i$, which means that $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$ is an anti-unifier of P_1 and P_2, and $(\Upsilon', M'', \widehat{\tau}'_1, \widehat{\tau}'_2)$ is an anti-unifier of N''_1 and N''_2. Then by the induction hypothesis, $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon_1, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ and $T \vDash N''_1 \overset{a}{\simeq} N''_2 \dashv (\Upsilon_2, M, \widehat{\tau}_3, \widehat{\tau}_4)$ succeed. The result of the algorithm is $(\Upsilon_1 \cup \Upsilon_2, Q \to M, \widehat{\tau}_1 \cup \widehat{\tau}_3, \widehat{\tau}_2 \cup \widehat{\tau}_4)$.

Case 6. $Q' = \widehat{\alpha}^+$ This case if not possible, since $T\,; \Upsilon' \vdash Q'$ means $\widehat{\alpha}^+ \in \Upsilon'$, but Υ' can only contain negative variables.

Case 7. Other positive cases are proved symmetrically to the corresponding negative ones.

□

Lemma 68 (Initiality of Anti-Unification).

+ *Assume that P_1 and P_2 are normalized, and $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$, then $(\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ is more specific than any other sound anti-unifier $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$, i.e. if*
  (1) $T\,; \Upsilon' \vdash Q'$,
  (2) $T\,;\cdot \vdash \widehat{\tau}'_i : \Upsilon'$ for $i \in \{1, 2\}$, and
  (3) $[\widehat{\tau}'_i]Q' = P_i$ for $i \in \{1, 2\}$
  then there exists $\widehat{\rho}$ such that $T\,; \Upsilon \vdash \widehat{\rho} : (\Upsilon'|_{\text{fv}\,Q'})$ and $[\widehat{\rho}]Q' = Q$. Moreover, $[\widehat{\rho}]\widehat{\beta}^-$ can be uniquely determined by $[\widehat{\tau}_1]\widehat{\beta}^-$, $[\widehat{\tau}_2]\widehat{\beta}^-$, and T.
− *Assume that N_1 and N_2 are normalized, and $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$, then $(\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$ is more specific than any other sound anti-unifier $(\Upsilon', M', \widehat{\tau}'_1, \widehat{\tau}'_2)$, i.e. if*
  (1) $T\,; \Upsilon' \vdash M'$,
  (2) $T\,;\cdot \vdash \widehat{\tau}'_i : \Upsilon'$ for $i \in \{1, 2\}$, and
  (3) $[\widehat{\tau}'_i]M' = N_i$ for $i \in \{1, 2\}$
  then there exists $\widehat{\rho}$ such that $T\,; \Upsilon \vdash \widehat{\rho} : (\Upsilon'|_{\text{fv}\,M'})$ and $[\widehat{\rho}]M' = M$. Moreover, $[\widehat{\rho}]\widehat{\beta}^-$ can be uniquely determined by $[\widehat{\tau}_1]\widehat{\beta}^-$, $[\widehat{\tau}_2]\widehat{\beta}^-$, and T.

Proof. First, let us assume that M' is a algorithmic variable $\widehat{\alpha}^-$. Then we can take $\widehat{\rho} = \widehat{\alpha}^- \mapsto M$, which satisfies the required properties:

- $T\,; \Upsilon \vdash \widehat{\rho} : (\Upsilon'|_{\text{fv}\,M'})$ holds since $\Upsilon'|_{\text{fv}\,M'} = \widehat{\alpha}^-$ and $T\,; \Upsilon \vdash M$ by the soundness of anti-unification (lemma 66);

- $[\widehat{\rho}]M' = M$ holds by construction
- $[\widehat{\rho}]\widehat{\alpha}^- = M$ is the anti-unifier of $N_1 = [\widehat{\tau}'_1]\widehat{\alpha}^-$ and $N_2 = [\widehat{\tau}'_2]\widehat{\alpha}^-$ in context T, and hence, it is uniquely determined by them (observation 5).

Now, we can assume that M' is not an algorithmic variable. We prove by induction on the derivation of $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ and mutually on the derivation of $T \vDash N_1 \overset{a}{\simeq} N_2 \dashv (\Upsilon, M, \widehat{\tau}_1, \widehat{\tau}_2)$.

Since M' is not a algorithmic variable, the substitution acting on M' preserves its outer constructor. In other words, $[\widehat{\tau}'_i]M' = N_i$ means that M', N_1 and N_2 have the same outer constructor. Let us consider the algorithmic anti-unification rule corresponding to this constructor, and show that it was successfully applied to anti-unify N_1 and N_2 (or P_1 and P_2).

Case 1. ($\text{Var}^{\overset{a}{\simeq}}_{-}$), i.e. $N_1 = \alpha^- = N_2$. This rule is applicable since it has no premises.
Then $\Upsilon = \cdot$, $M = \alpha^-$, and $\widehat{\tau}_1 = \widehat{\tau}_2 = \cdot$. Since $[\widehat{\tau}'_i]M' = N_i = \alpha^-$ and M' is not a algorithmic variable, $M' = \alpha^-$. Then we can take $\widehat{\rho} = \cdot$, which satisfies the required properties:

- $T\,;\Upsilon \vdash \widehat{\rho} : (\Upsilon'|_{\mathbf{fav}\,M'})$ holds vacuously since $\Upsilon'|_{\mathbf{fav}\,M'} = \emptyset$;
- $[\widehat{\rho}]M' = M$, that is $[\cdot]\alpha^- = \alpha^-$ holds by substitution properties;
- the unique determination of $[\widehat{\rho}]\widehat{\alpha}^-$ for $\widehat{\alpha}^- \in \Upsilon'|_{\mathbf{fav}\,M'} = \emptyset$ holds vacuously.

Case 2. ($\uparrow^{\overset{a}{\simeq}}$), i.e. $N_1 = \uparrow P_1$ and $N_2 = \uparrow P_2$.
Then since $[\widehat{\tau}'_i]M' = N_i = \uparrow P_i$ and M' is not a algorithmic variable, $M' = \uparrow Q'$, where $[\widehat{\tau}'_i]Q' = P_i$. Let us show that $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$ is an anti-unifier of P_1 and P_2.
(1) $T\,;\Upsilon' \vdash Q'$ holds by inversion of $T\,;\Upsilon' \vdash \uparrow Q'$;
(2) $T\,;\cdot \vdash \widehat{\tau}'_i : \Upsilon'$ holds by assumption;
(3) $[\widehat{\tau}'_i]Q' = P_i$ holds by assumption.
This way, by the completeness of anti-unification (lemma 67), the anti-unification algorithm succeeds on P_1 and P_2: $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$, which means that ($\uparrow^{\overset{a}{\simeq}}$) is applicable to infer $T \vDash \uparrow P_1 \overset{a}{\simeq} \uparrow P_2 \dashv (\Upsilon, \uparrow Q, \widehat{\tau}_1, \widehat{\tau}_2)$.
Moreover, by the induction hypothesis, $(\Upsilon, Q, \widehat{\tau}_1, \widehat{\tau}_2)$ is more specific than $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$, which immediately implies that $(\Upsilon, \uparrow Q, \widehat{\tau}_1, \widehat{\tau}_2)$ is more specific than $(\Upsilon', \uparrow Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$ (we keep the same $\widehat{\rho}$).

Case 3. ($\forall^{\overset{a}{\simeq}}$), i.e. $N_1 = \forall \overrightarrow{\alpha^+}.\, N'_1$ and $N_2 = \forall \overrightarrow{\alpha^+}.\, N'_2$. The proof is symmetric to the previous case. Notice that the context T is not changed in ($\forall^{\overset{a}{\simeq}}$), as it represents the context in which the anti-unification variables must be instantiated, rather than the context forming the types that are being anti-unified.

Case 4. ($\rightarrow^{\overset{a}{\simeq}}$), i.e. $N_1 = P_1 \rightarrow N'_1$ and $N_2 = P_2 \rightarrow N'_2$.
Then since $[\widehat{\tau}'_i]M' = N_i = P_i \rightarrow N'_i$ and M' is not a algorithmic variable, $M' = Q' \rightarrow M''$, where $[\widehat{\tau}'_i]Q' = P_i$ and $[\widehat{\tau}'_i]M'' = N''_i$.
Let us show that $(\Upsilon', Q', \widehat{\tau}'_1, \widehat{\tau}'_2)$ is an anti-unifier of P_1 and P_2.
(1) $T\,;\Upsilon' \vdash Q'$ holds by inversion of $T\,;\Upsilon' \vdash Q' \rightarrow M''$;
(2) $T\,;\cdot \vdash \widehat{\tau}'_i : \Upsilon'$ holds by assumption;
(3) $[\widehat{\tau}'_i]Q' = P_i$ holds by assumption.
Similarly, $(\Upsilon', M'', \widehat{\tau}'_1, \widehat{\tau}'_2)$ is an anti-unifier of N''_1 and N''_2.
Then by the completeness of anti-unification (lemma 67), the anti-unification algorithm succeeds on P_1 and P_2: $T \vDash P_1 \overset{a}{\simeq} P_2 \dashv (\Upsilon_1, Q, \widehat{\tau}_1, \widehat{\tau}_2)$; and on N'_1 and N'_2: $T \vDash N''_1 \overset{a}{\simeq} N''_2 \dashv (\Upsilon_2, M''', \widehat{\tau}_3, \widehat{\tau}_4)$. Notice that $\widehat{\tau}_1 \,\&\, \widehat{\tau}_3$ and $\widehat{\tau}_2 \,\&\, \widehat{\tau}_4$ are defined, in other words, for any $\widehat{\beta}^- \in \Upsilon_1 \cap \Upsilon_2$, $[\widehat{\tau}_1]\widehat{\beta}^- = [\widehat{\tau}_2]\widehat{\beta}^-$ and $[\widehat{\tau}_3]\widehat{\beta}^- = [\widehat{\tau}_4]\widehat{\beta}^-$, which follows immediately from

observation 6. This way, the algorithm proceeds by applying $(\overset{a}{\to\simeq})$ and returns $(\Upsilon_1 \cup \Upsilon_2, Q \to M''', \widehat{\tau}_1 \cup \widehat{\tau}_3, \widehat{\tau}_2 \cup \widehat{\tau}_4)$.
It is left to construct $\widehat{\rho}$ such that $T \,;\, \Upsilon \vdash \widehat{\rho} : (\Upsilon'|_{\mathbf{fav}\, M'})$ and $[\widehat{\rho}]M' = M$. By the induction hypothesis, there exist $\widehat{\rho}_1$ and $\widehat{\rho}_2$ such that $T \,;\, \Upsilon_1 \vdash \widehat{\rho}_1 : (\Upsilon'|_{\mathbf{fav}\, Q'})$, $T \,;\, \Upsilon_2 \vdash \widehat{\rho}_2 : (\Upsilon'|_{\mathbf{fav}\, M''})$, $[\widehat{\rho}_1]Q' = Q$, and $[\widehat{\rho}_2]M'' = M'''$.
Let us show that $\widehat{\rho} = \widehat{\rho}_1 \cup \widehat{\rho}_2$ satisfies the required properties:

- $T \,;\, \Upsilon_1 \cup \Upsilon_2 \vdash \widehat{\rho}_1 \cup \widehat{\rho}_2 : (\Upsilon'|_{\mathbf{fav}\, M'})$ holds since $\Upsilon'|_{\mathbf{fav}\, M'} = \Upsilon'|_{\mathbf{fav}\, Q' \to M''} = (\Upsilon'|_{\mathbf{fav}\, Q'}) \cup (\Upsilon'|_{\mathbf{fav}\, M''})$, $T \,;\, \Upsilon_1 \vdash \widehat{\rho}_1 : (\Upsilon'|_{\mathbf{fav}\, Q'})$ and $T \,;\, \Upsilon_2 \vdash \widehat{\rho}_2 : (\Upsilon'|_{\mathbf{fav}\, M''})$;
- $[\widehat{\rho}]M' = [\widehat{\rho}](Q' \to M'') = [\widehat{\rho}|_{\mathbf{fav}\, Q'}]Q' \to [\widehat{\rho}|_{\mathbf{fav}\, M''}]M'' = [\widehat{\rho}_1]Q' \to [\widehat{\rho}_2]M'' = Q \to M''' = M$;
- Since $[\widehat{\rho}]\widehat{\beta^-}$ is either equal to $[\widehat{\rho}_1]\widehat{\beta^-}$ or $[\widehat{\rho}_2]\widehat{\beta^-}$, it inherits their property that it is uniquely determined by $[\widehat{\tau}'_1]\widehat{\beta^-}$, $[\widehat{\tau}'_2]\widehat{\beta^-}$, and T.

Case 5. $P_1 = P_2 = \alpha^+$. This case is symmetric to case 1.
Case 6. $P_1 = {\downarrow}N_1$ and $P_2 = {\downarrow}N_2$. This case is symmetric to case 2
Case 7. $P_1 = \exists\overrightarrow{\alpha^-}.\, P'_1$ and $P_2 = \exists\overrightarrow{\alpha^-}.\, P'_2$. This case is symmetric to case 3

□

9.8 Upper Bounds

Observation 7 (Determinism of Least Upper Bound algorithm). *For types $T \vdash P_1$, and $T \vdash P_2$, if $T \vDash P_1 \vee P_2 = Q$ and $T \vDash P_1 \vee P_2 = Q'$ then $Q = Q'$.*

Proof. The shape of P_1 and P_2 uniquely determines the rule applied to infer the upper bound. By looking at the inference rules, it is easy to see that the result of the least upper bound algorithm depends on

- the inputs of the algorithm (that is P_1, P_2, and T), which are fixed;
- the result of the anti-unification algorithm applied to normalized input, which is deterministic by observation 5;
- the result of the recursive call, which is deterministic by the induction hypothesis.

□

Lemma 69 (Characterization of the Supertypes). *Let us define the set of upper bounds of a positive type* $\mathsf{UB}(P)$ *in the following way:*

$T \vdash P$	$\mathsf{UB}(T \vdash P)$
$T \vdash \beta^+$	$\{\exists\overrightarrow{\alpha^-}.\, \beta^+ \mid \textit{for } \overrightarrow{\alpha^-}\}$
$T \vdash \exists\overrightarrow{\beta^-}.\, Q$	$\mathsf{UB}(T, \overrightarrow{\beta^-} \vdash Q)$ *not using* $\overrightarrow{\beta^-}$
$T \vdash {\downarrow}M$	$\left\{ \exists\overrightarrow{\alpha^-}.\, {\downarrow}M' \;\middle|\; \begin{array}{l} \textit{for } \overrightarrow{\alpha^-}, M', \textit{and } \overrightarrow{N} \textit{ s.t.} \\ T \vdash N_i, T, \overrightarrow{\alpha^-} \vdash M', \textit{and } [\overrightarrow{N}/\overrightarrow{\alpha^-}]{\downarrow}M' \simeq^D {\downarrow}M \end{array} \right\}$

Then $\mathsf{UB}(T \vdash P) \equiv \{Q \mid T \vdash Q \geqslant P\}$.

Proof. By induction on $T \vdash P$.

Case 1. $P = \beta^+$
Immediately from lemma 17
Case 2. $P = \exists\overrightarrow{\beta^-}.\, P'$
Then if $T \vdash Q \geqslant \exists\overrightarrow{\beta^-}.\, P'$, then by lemma 16, $T, \overrightarrow{\beta^-} \vdash Q \geqslant P'$, and $\mathbf{fv}\, Q \cap \overrightarrow{\beta^-} = \emptyset$ by the

convention. The other direction holds by $(\exists^{\geqslant})$. This way, $\{Q \mid T \vdash Q \geqslant \exists\overrightarrow{\beta^-}.\, P'\} = \{Q \mid T, \overrightarrow{\beta^-} \vdash Q \geqslant P' \text{ s.t. } \mathbf{fv}\,(Q) \cap \overrightarrow{\beta^-} = \emptyset\}$. From the induction hypothesis, the latter is equal to $\mathsf{UB}(T, \overrightarrow{\beta^-} \vdash P')$ not using $\overrightarrow{\beta^-}$, i.e. $\mathsf{UB}(T \vdash \exists\overrightarrow{\beta^-}.\, P')$.

Case 3. $P = \downarrow M$

Then let us consider two subcases upper bounds without outer quantifiers (we denote the corresponding set restriction as $|_{\nexists}$) and upper bounds with outer quantifiers ($|_{\exists}$). We prove that for both of these groups, the restricted sets are equal.

a. $Q \neq \exists\overrightarrow{\beta^-}.\, Q'$

Then the last applied rule to infer $T \vdash Q \geqslant \downarrow M$ must be $(\downarrow^{\geqslant})$, which means $Q = \downarrow M'$, and by inversion, $T \vdash M' \simeq^{\leqslant} M$, then by lemma 32 and $(\downarrow^{\simeq^D})$, $\downarrow M' \simeq^D \downarrow M$. This way, $Q = \downarrow M' \in \{\downarrow M' \mid \downarrow M' \simeq^D \downarrow M\} = \mathsf{UB}(T \vdash \downarrow M)|_{\nexists}$.

In the other direction,
$$\begin{aligned} \downarrow M' \simeq^D \downarrow M &\Rightarrow T \vdash \downarrow M' \simeq^{\leqslant} \downarrow M && \text{by lemmas 26 and 27}\\ &\Rightarrow T \vdash \downarrow M' \geqslant \downarrow M && \text{by inversion}\end{aligned}$$

b. $Q = \exists\overrightarrow{\beta^-}.\, Q'$ (for non-empty $\overrightarrow{\beta^-}$)

Then the last rule applied to infer $T \vdash \exists\overrightarrow{\beta^-}.\, Q' \geqslant \downarrow M$ must be $(\exists^{\geqslant})$. Inversion of this rule gives us $T \vdash [\overrightarrow{N}/\overrightarrow{\beta^-}]Q' \geqslant \downarrow M$ for some $T \vdash N_i$. Notice that $[\overrightarrow{N}/\overrightarrow{\beta^-}]Q'$ has no outer quantifiers. Thus from case 3.*a*, $[\overrightarrow{N}/\overrightarrow{\beta^-}]Q' \simeq^D \downarrow M$, which is only possible if $Q' = \downarrow M'$. This way, $Q = \exists\overrightarrow{\beta^-}.\, \downarrow M' \in \mathsf{UB}(T \vdash \downarrow M)|_{\exists}$ (notice that $\overrightarrow{\beta^-}$ is not empty).

In the other direction,
$$\begin{aligned} [\overrightarrow{N}/\overrightarrow{\beta^-}]\downarrow M' \simeq^D \downarrow M &\Rightarrow T \vdash [\overrightarrow{N}/\overrightarrow{\beta^-}]\downarrow M' \simeq^{\leqslant} \downarrow M && \text{by lemmas 26 and 27}\\ &\Rightarrow T \vdash [\overrightarrow{N}/\overrightarrow{\beta^-}]\downarrow M' \geqslant \downarrow M && \text{by inversion}\\ &\Rightarrow T \vdash \exists\overrightarrow{\beta^-}.\, \downarrow M' \geqslant \downarrow M && \text{by } (\exists^{\geqslant})\end{aligned}$$

□

Lemma 70 (Characterization of the Normalized Supertypes). *For a normalized positive type $P = \mathbf{nf}\,(P)$, let us define the set of normalized upper bounds in the following way:*

$T \vdash P$	$\mathsf{NFUB}(T \vdash P)$
$T \vdash \beta^+$	$\{\beta^+\}$
$T \vdash \exists\overrightarrow{\beta^-}.\, P$	$\mathsf{NFUB}(T, \overrightarrow{\beta^-} \vdash P)$ *not using* $\overrightarrow{\beta^-}$
$T \vdash \downarrow M$	$\left\{ \exists\overrightarrow{\alpha^-}.\, \downarrow M' \;\middle|\; \begin{array}{l} \text{for } \overrightarrow{\alpha^-}, M', \text{and } \overrightarrow{N} \text{ s.t. } \mathbf{ord}\, \overrightarrow{\alpha^-} \,\mathbf{in}\, M' = \overrightarrow{\alpha^-},\\ T \vdash N_i, T, \overrightarrow{\alpha^-} \vdash M', \text{and } [\overrightarrow{N}/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow M \end{array} \right\}$

Then $\mathsf{NFUB}(T \vdash P) \equiv \{\mathbf{nf}\,(Q) \mid T \vdash Q \geqslant P\}$.

Proof. By induction on $T \vdash P$.

Case 1. $P = \beta^+$

Then from lemma 69, $\{\mathbf{nf}\,(Q) \mid T \vdash Q \geqslant \beta^+\} = \{\mathbf{nf}\,(\exists\overrightarrow{\alpha^-}.\, \beta^+) \mid \text{for some } \overrightarrow{\alpha^-}\} = \{\beta^+\}$

Case 2. $P = \exists \overrightarrow{\beta^-}. P'$

$\text{NFUB}(T \vdash \exists \overrightarrow{\beta^-}. P')$

$= \text{NFUB}(T, \overrightarrow{\beta^-} \vdash P')$ not using $\overrightarrow{\beta^-}$

$= \{\mathbf{nf}\,(Q) \mid T, \overrightarrow{\beta^-} \vdash Q \geqslant P'\}$ not using $\overrightarrow{\beta^-}$ by the induction hypothesis

$= \{\mathbf{nf}\,(Q) \mid T, \overrightarrow{\beta^-} \vdash Q \geqslant P' \text{ s.t. } \mathbf{fv}\, Q \cap \overrightarrow{\beta^-} = \emptyset\}$ $\mathbf{fv}\,\mathbf{nf}\,(Q) = \mathbf{fv}\, Q$ by lemma 38

$= \{\mathbf{nf}\,(Q) \mid Q \in \text{UB}(T, \overrightarrow{\beta^-} \vdash P') \text{ s.t. } \mathbf{fv}\, Q \cap \overrightarrow{\beta^-} = \emptyset\}$ by lemma 69

$= \{\mathbf{nf}\,(Q) \mid Q \in \text{UB}(T \vdash \exists \overrightarrow{\beta^-}. P')\}$ by the definition of UB

$= \{\mathbf{nf}\,(Q) \mid T \vdash Q \geqslant \exists \overrightarrow{\beta^-}. P'\}$ by lemma 69

Case 3. $P = \downarrow M$ Let us prove the set equality by two inclusions.

$\subseteq$ Suppose that $T \vdash Q \geqslant \downarrow M$ and M is normalized.
By lemma 69, $Q \in \text{UB}(T \vdash \downarrow M)$. Then by definition of UB, $Q = \exists \overrightarrow{\alpha^-}. \downarrow M'$ for some $\overrightarrow{\alpha^-}$, M', and $T \vdash \sigma : \overrightarrow{\alpha^-}$ s.t. $[\sigma]\downarrow M' \simeq^D \downarrow M$.
We need to show that $\mathbf{nf}\,(Q) \in \text{NFUB}(T \vdash \downarrow M)$. Notice that $\mathbf{nf}\,(Q) = \mathbf{nf}\,(\exists \overrightarrow{\alpha^-}. \downarrow M') = \exists \overrightarrow{\alpha^-}_0. \downarrow M_0$, where $\mathbf{nf}\,(M') = M_0$ and $\mathbf{ord}\, \overrightarrow{\alpha^-}\, \mathbf{in}\, M_0 = \overrightarrow{\alpha^-}_0$.
The belonging of $\exists \overrightarrow{\alpha^-}_0. \downarrow M_0$ to $\text{NFUB}(T \vdash \downarrow M)$ means that

(1) $\mathbf{ord}\, \overrightarrow{\alpha^-}_0\, \mathbf{in}\, M_0 = \overrightarrow{\alpha^-}_0$ and
(2) that there exists $T \vdash \sigma_0 : \overrightarrow{\alpha^-}_0$ such that $[\sigma_0]\downarrow M_0 = \downarrow M$.

The first requirement holds by corollary 13. To show the second requirement, we construct σ_0 as $\mathbf{nf}\,(\sigma|_{\mathbf{fv}\, M'})$. Let us show the required properties of σ_0:

(1) $T \vdash \sigma_0 : \overrightarrow{\alpha^-}_0$. Notice that by lemma 5, $T \vdash \sigma|_{\mathbf{fv}\,(M')} : \overrightarrow{\alpha^-} \cap \mathbf{fv}\,(M')$, which we rewrite as $T \vdash \sigma|_{\mathbf{fv}\,(M')} : \overrightarrow{\alpha^-}_0$ (since by lemma 33 $\overrightarrow{\alpha^-}_0 = \overrightarrow{\alpha^-} \cap \mathbf{fv}\, M_0$ as sets, and $\mathbf{fv}\,(M_0) = \mathbf{fv}\,(M')$ by lemma 38). Then by lemma 40, $T \vdash \mathbf{nf}\,(\sigma|_{\mathbf{fv}\,(M')}) : \overrightarrow{\alpha^-}_0$, that is $T \vdash \sigma_0 : \overrightarrow{\alpha^-}_0$.
(2) $[\sigma_0]\downarrow M_0 = \downarrow M$. $[\sigma]\downarrow M' \simeq^D \downarrow M$ means $[\sigma|_{\mathbf{fv}\,(M')}]\downarrow M' \simeq^D \downarrow M$ by lemma 4. Then by lemma 43, $\mathbf{nf}\,([\sigma|_{\mathbf{fv}\,(M')}]\downarrow M') = \mathbf{nf}\,(\downarrow M)$, implying $[\sigma_0]\downarrow M_0 = \mathbf{nf}\,(\downarrow M)$ by lemma 41, and further $[\sigma_0]\downarrow M_0 = \downarrow M$ by lemma 44 (since $\downarrow M$ is normal by assumption).

$\supseteq$ Suppose that a type belongs to $\text{NFUB}(T \vdash \downarrow M)$ for a normalized $\downarrow M$. Then it must have shape $\exists \overrightarrow{\alpha^-}_0. \downarrow M_0$ for some $\overrightarrow{\alpha^-}_0$, M_0, and $T \vdash \sigma_0 : \overrightarrow{\alpha^-}_0$ such that $\mathbf{ord}\, \overrightarrow{\alpha^-}_0\, \mathbf{in}\, M_0 = \overrightarrow{\alpha^-}_0$ and $[\sigma_0]\downarrow M_0 = \downarrow M$. It suffices to show that (1) $\exists \overrightarrow{\alpha^-}_0. \downarrow M_0$ is normalized itself, and (2) $T \vdash \exists \overrightarrow{\alpha^-}_0. \downarrow M_0 \geqslant \downarrow M$.

(1) By definition, $\mathbf{nf}\,(\exists \overrightarrow{\alpha^-}_0. \downarrow M_0) = \exists \overrightarrow{\alpha^-}_1. \downarrow M_1$, where $M_1 = \mathbf{nf}\,(M_0)$ and $\mathbf{ord}\, \overrightarrow{\alpha^-}_0\, \mathbf{in}\, M_1 = \overrightarrow{\alpha^-}_1$. First, notice that by lemmas 37 and 39, $\mathbf{ord}\, \overrightarrow{\alpha^-}_0\, \mathbf{in}\, M_1 = \mathbf{ord}\, \overrightarrow{\alpha^-}_0\, \mathbf{in}\, M_0 = \overrightarrow{\alpha^-}_0$. This way, $\mathbf{nf}\,(\exists \overrightarrow{\alpha^-}_0. \downarrow M_0) = \exists \overrightarrow{\alpha^-}_0. \downarrow \mathbf{nf}\,(M_0)$. Second, M_0 is normalized by lemma 45, since $[\sigma_0]\downarrow M_0 = \downarrow M$ is normal. As such, $\mathbf{nf}\,(\exists \overrightarrow{\alpha^-}_0. \downarrow M_0) = \exists \overrightarrow{\alpha^-}_0. \downarrow M_0$, in other words, $\exists \overrightarrow{\alpha^-}_0. \downarrow M_0$ is normalized.
(2) $T \vdash \exists \overrightarrow{\alpha^-}_0. \downarrow M_0 \geqslant \downarrow M$ holds immediately by $(\exists^{\geqslant})$ with the substitution σ_0. Notice that $T \vdash [\sigma_0]\downarrow M_0 \geqslant \downarrow M$ follows from $[\sigma_0]\downarrow M_0 = \downarrow M$ by reflexivity of subtyping (lemma 20).

□

Lemma 71. *Upper bounds of a type do not depend on the context as soon as the type is well-formed in it.*
If $T_1 \vdash P$ *and* $T_2 \vdash P$ *then* $\text{UB}(T_1 \vdash P) = \text{UB}(T_2 \vdash P)$ *and* $\text{NFUB}(T_1 \vdash P) = \text{NFUB}(T_2 \vdash P)$

Proof. We prove both inclusions by structural induction on P.

Case 1. $P = \beta^+$ Then $\mathsf{UB}(T_1 \vdash \beta^+) = \mathsf{UB}(T_2 \vdash \beta^+) = \{\exists\overrightarrow{\alpha^-}.\,\beta^+ \mid \text{for some } \overrightarrow{\alpha^-}\}$. $\mathsf{NFUB}(T_1 \vdash \beta^+) = \mathsf{NFUB}(T_2 \vdash \beta^+) = \{\beta^+\}$.

Case 2. $P = \exists\overrightarrow{\beta^-}.\,P'$. Then $\mathsf{UB}(T_1 \vdash \exists\overrightarrow{\beta^-}.\,P') = \mathsf{UB}(T_1, \overrightarrow{\beta^-} \vdash P')$ not using $\overrightarrow{\beta^-}$. $\mathsf{UB}(T_2 \vdash \exists\overrightarrow{\beta^-}.\,P') = \mathsf{UB}(T_2, \overrightarrow{\beta^-} \vdash P')$ not using $\overrightarrow{\beta^-}$. By the induction hypothesis, $\mathsf{UB}(T_1, \overrightarrow{\beta^-} \vdash P') = \mathsf{UB}(T_2, \overrightarrow{\beta^-} \vdash P')$, and if we restrict these sets to the same domain, they stay equal. Analogously, $\mathsf{NFUB}(T_1 \vdash \exists\overrightarrow{\beta^-}.\,P') = \mathsf{NFUB}(T_2 \vdash \exists\overrightarrow{\beta^-}.\,P')$.

Case 3. $P = \downarrow M$. Suppose that $\exists\overrightarrow{\alpha^-}.\downarrow M' \in \mathsf{UB}(T_1 \vdash \downarrow M)$. It means that $T_1, \overrightarrow{\alpha^-} \vdash M'$ and there exist $T_1 \vdash \overrightarrow{N}$ s.t. $[\overrightarrow{N}/\overrightarrow{\alpha^-}]\downarrow M' \simeq^D \downarrow M$, or in other terms, there exists $T_1 \vdash \sigma : \overrightarrow{\alpha^-}$ such that $[\sigma]\downarrow M' \simeq^D \downarrow M$.

We need to show that $\exists\overrightarrow{\alpha^-}.\downarrow M' \in UB(T_2 \vdash \downarrow M)$, in other words, $T_2, \overrightarrow{\alpha^-} \vdash M'$ and there exists $T_2 \vdash \sigma_0 : \overrightarrow{\alpha^-}$ such that $[\sigma_0]\downarrow M' \simeq^D \downarrow M$.

First, let us show $T_2, \overrightarrow{\alpha^-} \vdash M'$. Notice that $[\sigma]\downarrow M' \simeq^D \downarrow M$ implies $\mathbf{fv}([\sigma]M') = \mathbf{fv}(\downarrow M)$ by lemma 24. By lemma 13, $\mathbf{fv}(M') \setminus \overrightarrow{\alpha^-} \subseteq \mathbf{fv}([\sigma]M')$. This way, $\mathbf{fv}(M') \setminus \overrightarrow{\alpha^-} \subseteq \mathbf{fv}(M)$, implying $\mathbf{fv}(M') \subseteq \mathbf{fv}(M) \cup \overrightarrow{\alpha^-}$. By lemma 1, $T_2 \vdash \downarrow M$ implies $\mathbf{fv}\,M \subseteq T_2$, hence, $\mathbf{fv}\,M' \subseteq (T_2, \overrightarrow{\alpha^-})$, which by corollary 1 means $T_2, \overrightarrow{\alpha^-} \vdash M'$.

Second, let us construct the required σ_0 in the following way:

$$\begin{cases} [\sigma_0]\alpha_i^- = [\sigma]\alpha_i^- & \text{for } \alpha_i^- \in \overrightarrow{\alpha^-} \cap \mathbf{fv}(M') \\ [\sigma_0]\alpha_i^- = \forall\gamma^+.\uparrow\gamma^+ & \text{for } \alpha_i^- \in \overrightarrow{\alpha^-} \setminus \mathbf{fv}(M') \\ [\sigma_0]\gamma^\pm = \gamma^\pm & \text{for any other } \gamma^\pm \end{cases}$$

This construction of a substitution coincides with the one from the proof of lemma 18. This way, for σ_0, hold the same properties:

(1) $[\sigma_0]M' = [\sigma]M'$, which in particular, implies $[\sigma_0]\downarrow M = [\sigma]\downarrow M$, and thus, $[\sigma]\downarrow M' \simeq^D \downarrow M$ can be rewritten to $[\sigma_0]\downarrow M' \simeq^D \downarrow M$; and

(2) $\mathbf{fv}([\sigma]M') \vdash \sigma_0 : \overrightarrow{\alpha^-}$, which, as noted above, can be rewritten to $\mathbf{fv}(M) \vdash \sigma_0 : \overrightarrow{\alpha^-}$, and since $\mathbf{fv}\,M \subseteq T_2$, weakened to $T_2 \vdash \sigma_0 : \overrightarrow{\alpha^-}$.

The proof of $\mathsf{NFUB}(T_1 \vdash \downarrow M) \subseteq \mathsf{NFUB}(T_2 \vdash \downarrow M)$ is analogous. The differences are:

(1) $\mathbf{ord}\,\overrightarrow{\alpha^-}\,\mathbf{in}\,M' = \overrightarrow{\alpha^-}$ holds by assumption,

(2) $[\sigma]\downarrow M' = \downarrow M$ implies $\mathbf{fv}([\sigma]M') = \mathbf{fv}(\downarrow M)$ by rewriting,

(3) $[\sigma]\downarrow M' = \downarrow M$ and $[\sigma_0]\downarrow M = [\sigma]\downarrow M$ imply $[\sigma_0]\downarrow M' = \downarrow M$ by rewriting.

□

Lemma 72 (Soundness of the Least Upper Bound). *For types $T \vdash P_1$, and $T \vdash P_2$, if $T \vDash P_1 \vee P_2 = Q$ then*

(i) $T \vdash Q$

(ii) $T \vdash Q \geqslant P_1$ *and* $T \vdash Q \geqslant P_2$

Proof. Induction on $T \vDash P_1 \vee P_2 = Q$.

Case 1. $T \vDash \alpha^+ \vee \alpha^+ = \alpha^+$

Then $T \vdash \alpha^+$ by assumption, and $T \vdash \alpha^+ \geqslant \alpha^+$ by ($\textsc{Var}_+^{\geqslant}$).

Case 2. $T \vDash \exists\overrightarrow{\alpha^-}.\,P_1 \vee \exists\overrightarrow{\beta^-}.\,P_2 = Q$

Then by inversion of $T \vdash \exists\overrightarrow{\alpha^-}.\,P_i$ and weakening, $T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-} \vdash P_i$, hence, the induction hypothesis applies to $T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-} \vDash P_1 \vee P_2 = Q$. Then

(i) $T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-} \vdash Q$,

(ii) $T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-} \vdash Q \geqslant P_1$,

(iii) $T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-} \vdash Q \geqslant P_2$.

To prove $T \vdash Q$, it suffices to show that $\mathbf{fv}\,(Q) \cap (T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-}) = \mathbf{fv}\,(Q) \cap T$ (and then apply lemma 2). The inclusion right-to-left is self-evident. To show $\mathbf{fv}\,(Q) \cap (T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-}) \subseteq \mathbf{fv}\,(Q) \cap T$, we prove that $\mathbf{fv}\,(Q) \subseteq T$.

$\mathbf{fv}\,(Q) \subseteq \mathbf{fv}\,P_1 \cap \mathbf{fv}\,P_2$ by lemma 15

$\subseteq ((T, \overrightarrow{\alpha^-}) \setminus \overrightarrow{\beta^-}) \cap ((T, \overrightarrow{\beta^-}) \setminus \overrightarrow{\alpha^-})$ since $T \vdash \exists \overrightarrow{\alpha^-}.\, P_1$, $\mathbf{fv}\,(P_1) \subseteq (T, \overrightarrow{\alpha^-}) = (T, \overrightarrow{\alpha^-}) \setminus \overrightarrow{\beta^-}$(the latter is because by the Barendregt's convention, $(T, \overrightarrow{\alpha^-}) \cap \overrightarrow{\beta^-} = \emptyset$) similarly, $\mathbf{fv}\,(P_2) \subseteq (T, \overrightarrow{\beta^-}) \setminus \overrightarrow{\alpha^-}$

$\subseteq T$

To show $T \vdash Q \geqslant \exists \overrightarrow{\alpha^-}.\, P_1$, we apply ($\exists^{\geqslant}$). Then $T, \overrightarrow{\alpha^-} \vdash Q \geqslant P_1$ holds since $T, \overrightarrow{\alpha^-}, \overrightarrow{\beta^-} \vdash Q \geqslant P_1$ (by the induction hypothesis), $T, \overrightarrow{\alpha^-} \vdash Q$ (by weakening), and $T, \overrightarrow{\alpha^-} \vdash P_1$.
Judgment $T \vdash Q \geqslant \exists \overrightarrow{\beta^-}.\, P_2$ is proved symmetrically.

Case 3. $T \vDash {\downarrow}N \vee {\downarrow}M = \exists \overrightarrow{\alpha^-}.\, [\overrightarrow{\alpha^-}/\Upsilon]P$. By the inversion, $T, \cdot \vDash \mathbf{nf}\,({\downarrow}N) \overset{a}{\simeq} \mathbf{nf}\,({\downarrow}M) \dashv (\Upsilon, P, \widehat{\tau}_1, \widehat{\tau}_2)$. Then by the soundness of anti-unification (lemma 66),

(i) $T\,;\Upsilon \vdash P$, then by lemma 53,

$$T, \overrightarrow{\alpha^-} \vdash [\overrightarrow{\alpha^-}/\Upsilon]P \tag{7}$$

(ii) $T\,;\cdot \vdash \widehat{\tau}_1 : \Upsilon$ and $T\,;\cdot \vdash \widehat{\tau}_2 : \Upsilon$. Assuming that $\Upsilon = \widehat{\beta}^-_1, .., \widehat{\beta}^-_n$, the antiunification solutions $\widehat{\tau}_1$ and $\widehat{\tau}_2$ can be put explicitly as $\widehat{\tau}_1 = (\widehat{\beta}^-_1 :\simeq N_1, .., \widehat{\beta}^-_n :\simeq N_n)$, and $\widehat{\tau}_2 = (\widehat{\beta}^-_1 :\simeq M_1, .., \widehat{\beta}^-_n :\simeq M_n)$. Then

$$\widehat{\tau}_1 = (\overrightarrow{N}/\overrightarrow{\alpha^-}) \circ (\overrightarrow{\alpha^-}/\Upsilon) \tag{8}$$

$$\widehat{\tau}_2 = (\overrightarrow{M}/\overrightarrow{\alpha^-}) \circ (\overrightarrow{\alpha^-}/\Upsilon) \tag{9}$$

(iii) $[\widehat{\tau}_1]Q = P_1$ and $[\widehat{\tau}_2]Q = P_1$, which, by 8 and 9, means

$$[\overrightarrow{N}/\overrightarrow{\alpha^-}][\overrightarrow{\alpha^-}/\Upsilon]P = \mathbf{nf}\,({\downarrow}N) \tag{10}$$

$$[\overrightarrow{M}/\overrightarrow{\alpha^-}][\overrightarrow{\alpha^-}/\Upsilon]P = \mathbf{nf}\,({\downarrow}M) \tag{11}$$

Then $T \vdash \exists \overrightarrow{\alpha^-}.\, [\overrightarrow{\alpha^-}/\Upsilon]P$ follows directly from 7.
To show $T \vdash \exists \overrightarrow{\alpha^-}.\, [\overrightarrow{\alpha^-}/\Upsilon]P \geqslant {\downarrow}N$, we apply ($\exists^{\geqslant}$), instantiating $\overrightarrow{\alpha^-}$ with $\overrightarrow{N}$. Then $T \vdash [\overrightarrow{N}/\overrightarrow{\alpha^-}][\overrightarrow{\alpha^-}/\Upsilon]P \geqslant {\downarrow}N$ follows from 10 and since $T \vdash \mathbf{nf}\,({\downarrow}N) \geqslant {\downarrow}N$ (by corollary 16).
Analogously, instantiating $\overrightarrow{\alpha^-}$ with $\overrightarrow{M}$, gives us $T \vdash [\overrightarrow{M}/\overrightarrow{\alpha^-}][\overrightarrow{\alpha^-}/\Upsilon]P \geqslant {\downarrow}M$ (from 11), and hence, $T \vdash \exists \overrightarrow{\alpha^-}.\, [\overrightarrow{\alpha^-}/\Upsilon]P \geqslant {\downarrow}M$.

□

Lemma 73 (Completeness and Initiality of the Least Upper Bound). *For types $T \vdash P_1$, $T \vdash P_2$, and $T \vdash Q$ such that $T \vdash Q \geqslant P_1$ and $T \vdash Q \geqslant P_2$, there exists Q' s.t. $T \vDash P_1 \vee P_2 = Q'$ and $T \vdash Q \geqslant Q'$.*

Proof. Induction on the pair (P_1, P_2). From lemma 70, $Q \in \mathrm{UB}(T \vdash P_1) \cap \mathrm{UB}(T \vdash P_2)$. Let us consider the cases of what P_1 and P_2 are (i.e. the last rules to infer $T \vdash P_i$).

Case 1. $P_1 = \exists \overrightarrow{\beta^-}_1.\, Q_1$, $P_2 = \exists \overrightarrow{\beta^-}_2.\, Q_2$, where either $\overrightarrow{\beta^-}_1$ or $\overrightarrow{\beta^-}_2$ is not empty

Then

$$\begin{aligned} Q &\in \mathrm{UB}(T \vdash \exists \overrightarrow{\beta^-}_1.\, Q_1) \cap \mathrm{UB}(T \vdash \exists \overrightarrow{\beta^-}_2.\, Q_2) \\ &\subseteq \mathrm{UB}(T, \overrightarrow{\beta^-}_1 \vdash Q_1) \cap \mathrm{UB}(T, \overrightarrow{\beta^-}_2 \vdash Q_2) && \text{definition of UB} \\ &= \mathrm{UB}(T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vdash Q_1) \cap \mathrm{UB}(T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vdash Q_2) && \text{by lemma 71} \\ &= \{Q' \mid T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vdash Q' \geqslant Q_1\} \cap \{Q' \mid T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vdash Q' \geqslant Q_2\} && \text{by lemma 69} \end{aligned}$$

It means that $T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vdash Q \geqslant Q_1$ and $T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vdash Q \geqslant Q_2$. Then the next step of the algorithm—the recursive call $T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vDash Q_1 \vee Q_2 = Q'$ terminates by the induction hypothesis, and moreover, $T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vdash Q \geqslant Q'$. This way, the result of the algorithm is Q', i.e. $T \vDash P_1 \vee P_2 = Q'$.

Since both Q and Q' are sound upper bounds, $T \vdash Q$ and $T \vdash Q'$, and therefore, $T, \overrightarrow{\beta^-}_1, \overrightarrow{\beta^-}_2 \vdash Q \geqslant Q'$ can be strengthened to $T \vdash Q \geqslant Q'$ by lemma 18.

Case 2. $P_1 = \alpha^+$ and $P_2 = \downarrow N$

Then the set of common upper bounds of $\downarrow N$ and α^+ is empty, and thus, $Q \in \mathrm{UB}(T \vdash P_1) \cap \mathrm{UB}(T \vdash P_2)$ gives a contradiction:

$$\begin{aligned} Q &\in \mathrm{UB}(T \vdash \alpha^+) \cap \mathrm{UB}(T \vdash \downarrow N) \\ &= \{\exists \overrightarrow{\alpha^-}.\, \alpha^+ \mid \cdots\} \cap \{\exists \overrightarrow{\beta^-}.\, \downarrow M' \mid \cdots\} && \text{by the definition of UB} \\ &= \emptyset && \text{since } \alpha^+ \neq \downarrow M' \text{ for any } M' \end{aligned}$$

Case 3. $P_1 = \downarrow N$ and $P_2 = \alpha^+$

Symmetric to case 2

Case 4. $P_1 = \alpha^+$ and $P_2 = \beta^+$ (where $\beta^+ \neq \alpha^+$)

Similarly to case 2, the set of common upper bounds is empty, which leads to the contradiction:

$$\begin{aligned} Q &\in \mathrm{UB}(T \vdash \alpha^+) \cap \mathrm{UB}(T \vdash \beta^+) \\ &= \{\exists \overrightarrow{\alpha^-}.\, \alpha^+ \mid \cdots\} \cap \{\exists \overrightarrow{\beta^-}.\, \beta^+ \mid \cdots\} && \text{by the definition of UB} \\ &= \emptyset && \text{since } \alpha^+ \neq \beta^+ \end{aligned}$$

Case 5. $P_1 = \alpha^+$ and $P_2 = \alpha^+$

Then the algorithm terminates in one step ((Var$^\vee$)) and the result is α^+, i.e. $T \vDash \alpha^+ \vee \alpha^+ = \alpha^+$. Since $Q \in \mathrm{UB}(T \vdash \alpha^+)$, $Q = \exists \overrightarrow{\alpha^-}.\, \alpha^+$. Then $T \vdash \exists \overrightarrow{\alpha^-}.\, \alpha^+ \geqslant \alpha^+$ by $(\exists^\geqslant)$: $\overrightarrow{\alpha^-}$ can be instantiated with arbitrary negative types (for example $\forall \beta^+.\, \uparrow \beta^+$), since the substitution for unused variables does not change the term $[\overrightarrow{N}/\overrightarrow{\alpha^-}]\alpha^+ = \alpha^+$, and then $T \vdash \alpha^+ \geqslant \alpha^+$ by (Var$^\geqslant_+$).

Case 6. $P_1 = \downarrow M_1$ and $P_2 = \downarrow M_2$

Then on the next step, the algorithm tries to anti-unify **nf** $(\downarrow M_1)$ and **nf** $(\downarrow M_2)$. By lemma 67, to show that the anti-unification algorithm terminates, it suffices to demonstrate that a sound anti-unification solution exists.

Notice that

$$\begin{aligned}
&\mathbf{nf}\,(Q) \in \mathsf{NFUB}(T \vdash \mathbf{nf}\,(\downarrow M_1)) \cap \mathsf{NFUB}(T \vdash \mathbf{nf}\,(\downarrow M_2))\\
&\qquad = \mathsf{NFUB}(T \vdash \downarrow\mathbf{nf}\,(M_1)) \cap \mathsf{NFUB}(T \vdash \downarrow\mathbf{nf}\,(M_2))\\
&\qquad\quad \left\{ \exists\overrightarrow{\alpha^-}.\, \downarrow M' \;\middle|\; \begin{array}{l} \text{for } \overrightarrow{\alpha^-}, M', \text{and } \overrightarrow{N} \text{ s.t. } \mathbf{ord}\,\overrightarrow{\alpha^-}\,\mathbf{in}\,M' = \overrightarrow{\alpha^-},\\ T \vdash N_i,\, T, \overrightarrow{\alpha^-} \vdash M', \text{ and } [\overrightarrow{N}/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow\mathbf{nf}\,(M_1) \end{array} \right\}\\
&\qquad = \;\cap\\
&\qquad\quad \left\{ \exists\overrightarrow{\alpha^-}.\, \downarrow M' \;\middle|\; \begin{array}{l} \text{for } \overrightarrow{\alpha^-}, M', \text{and } \overrightarrow{N} \text{ s.t. } \mathbf{ord}\,\overrightarrow{\alpha^-}\,\mathbf{in}\,M' = \overrightarrow{\alpha^-},\\ T \vdash \overrightarrow{N}_1, T \vdash \overrightarrow{N}_2, T, \overrightarrow{\alpha^-} \vdash M', \text{ and } [\overrightarrow{N}/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow\mathbf{nf}\,(M_2) \end{array} \right\}\\
&\qquad = \left\{ \exists\overrightarrow{\alpha^-}.\, \downarrow M' \;\middle|\; \begin{array}{l} \text{for } \overrightarrow{\alpha^-}, M', \overrightarrow{N}_1 \text{ and } \overrightarrow{N}_2 \text{ s.t. } \mathbf{ord}\,\overrightarrow{\alpha^-}\,\mathbf{in}\,M' = \overrightarrow{\alpha^-},\\ T \vdash \overrightarrow{N}_1, T \vdash \overrightarrow{N}_2, T, \overrightarrow{\alpha^-} \vdash M', [\overrightarrow{N}_1/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow\mathbf{nf}\,(M_1)\\ \text{, and } [\overrightarrow{N}_2/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow\mathbf{nf}\,(M_2) \end{array} \right\}
\end{aligned}$$

The fact that the latter set is non-empty means that there exist $\overrightarrow{\alpha^-}, M', \overrightarrow{N}_1$ and $\overrightarrow{N}_2$ such that

(i) $T, \overrightarrow{\alpha^-} \vdash M'$ (notice that M' is normal)

(ii) $T \vdash \overrightarrow{N}_1$ and $T \vdash \overrightarrow{N}_1$,

(iii) $[\overrightarrow{N}_1/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow\mathbf{nf}\,(M_1)$ and $[\overrightarrow{N}_2/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow\mathbf{nf}\,(M_2)$

For each negative variable α^- from $\overrightarrow{\alpha^-}$, let us choose a fresh negative anti-unification variable $\widehat{\alpha}^-$, and denote the list of these variables as $\overrightarrow{\widehat{\alpha}^-}$. Let us show that $(\overrightarrow{\widehat{\alpha}^-},\ [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\downarrow M',\ \overrightarrow{N}_1/\overrightarrow{\widehat{\alpha}^-},\ \overrightarrow{N}_2/\overrightarrow{\widehat{\alpha}^-})$ is a sound anti-unifier of $\mathbf{nf}\,(\downarrow M_1)$ and $\mathbf{nf}\,(\downarrow M_2)$ in context T:

- $\overrightarrow{\widehat{\alpha}^-}$ is negative by construction,
- $T\,;\overrightarrow{\widehat{\alpha}^-} \vdash [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\downarrow M'$ because $T, \overrightarrow{\alpha^-} \vdash \downarrow M'$ (lemma 52),
- $T\,;\cdot \vdash (\overrightarrow{N}_1/\overrightarrow{\widehat{\alpha}^-}) : \overrightarrow{\widehat{\alpha}^-}$ because $T \vdash \overrightarrow{N}_1$ and $T\,;\cdot \vdash (\overrightarrow{N}_2/\overrightarrow{\widehat{\alpha}^-}) : \overrightarrow{\widehat{\alpha}^-}$ because $T \vdash \overrightarrow{N}_2$,
- $[\overrightarrow{N}_1/\overrightarrow{\widehat{\alpha}^-}][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\downarrow M' = [\overrightarrow{N}_1/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow\mathbf{nf}\,(M_1) = \mathbf{nf}\,(\downarrow M_1)$.
- $[\overrightarrow{N}_2/\overrightarrow{\widehat{\alpha}^-}][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\downarrow M' = [\overrightarrow{N}_2/\overrightarrow{\alpha^-}]\downarrow M' = \downarrow\mathbf{nf}\,(M_2) = \mathbf{nf}\,(\downarrow M_2)$.

Then by the completeness of the anti-unification (lemma 67), the anti-unification algorithm terminates, so is the Least Upper Bound algorithm invoking it, i.e. $Q' = \exists\overrightarrow{\beta^-}.\ [\overrightarrow{\beta^-}/\Upsilon]\,P$, where $(\Upsilon, P, \widehat{\tau}_1, \widehat{\tau}_2)$ is the result of the anti-unification of $\mathbf{nf}\,(\downarrow M_1)$ and $\mathbf{nf}\,(\downarrow M_2)$ in context T.

Moreover, lemma 67 also says that the found anti-unification solution is initial, i.e. there exists $\widehat{\tau}$ such that $T\,;\Upsilon \vdash \widehat{\tau} : \overrightarrow{\widehat{\alpha}^-}$ and $[\widehat{\tau}][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\downarrow M' = P$.

Let σ be a sequential Kleisli composition of the following substitutions: (i) $\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}$, (ii) $\widehat{\tau}$, and (iii) $\overrightarrow{\beta^-}/\Upsilon$. Notice that $T, \overrightarrow{\beta^-} \vdash \sigma : \overrightarrow{\alpha^-}$ and $[\sigma]\downarrow M' = [\overrightarrow{\beta^-}/\Upsilon][\widehat{\tau}][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\downarrow M' = [\overrightarrow{\beta^-}/\Upsilon]\,P$. In particular, from the reflexivity of subtyping: $T, \overrightarrow{\beta^-} \vdash [\sigma]\downarrow M' \geqslant [\overrightarrow{\beta^-}/\Upsilon]\,P$.

It allows us to show $T \vdash \mathbf{nf}\,(Q) \geqslant Q'$, i.e. $T \vdash \exists\overrightarrow{\alpha^-}.\downarrow M' \geqslant \exists\overrightarrow{\beta^-}.\ [\overrightarrow{\beta^-}/\Upsilon]\,P$, by applying $(\exists^{\geqslant})$, instantiating $\overrightarrow{\alpha^-}$ with respect to σ. Finally, $T \vdash Q \geqslant Q'$ by transitively combining $T \vdash \mathbf{nf}\,(Q) \geqslant Q'$ and $T \vdash Q \geqslant \mathbf{nf}\,(Q)$ (holds by corollary 16 and inversion).

□

9.9 Upgrade

Let us consider a type P well-formed in T. Some of its T-supertypes are also well-formed in a smaller context $\Theta \subseteq T$. The upgrade is the operation that returns the least of such supertypes.

Observation 8 (Upgrade determinism). *Assuming P is well-formed in $T \subseteq \Theta$, if* **upgrade** $T \vdash P$ **to** $\Theta = Q$ *and* **upgrade** $T \vdash P$ **to** $\Theta = Q'$ *are defined then $Q = Q'$.*

Proof. It follows directly from observation 7, and the convention that the fresh variables are chosen by a fixed deterministic algorithm (section 2.2). □

Lemma 74 (Soundness of Upgrade). *Assuming P is well-formed in $T = \Theta, \overrightarrow{\alpha^{\pm}}$, if* $\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q$ *then*

(1) $\Theta \vdash Q$
(2) $T \vdash Q \geqslant P$

Lemma 95 (Soundness of Upgrade). *Assuming P is well-formed in $T = \Theta, \overrightarrow{\alpha^{\pm}}$, if* $\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q$ *then*

(1) $\Theta \vdash Q$
(2) $T \vdash Q \geqslant P$

Proof. By inversion, $\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q$ means that for fresh $\overrightarrow{\beta^{\pm}}$ and $\overrightarrow{\gamma^{\pm}}$, $\Theta, \overrightarrow{\beta^{\pm}}, \overrightarrow{\gamma^{\pm}} \vDash [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P \vee [\overrightarrow{\gamma^{\pm}}/\overrightarrow{\alpha^{\pm}}]P = Q$. Then by the soundness of the least upper bound (lemma 72),

(1) $\Theta, \overrightarrow{\beta^{\pm}}, \overrightarrow{\gamma^{\pm}} \vdash Q$,
(2) $\Theta, \overrightarrow{\beta^{\pm}}, \overrightarrow{\gamma^{\pm}} \vdash Q \geqslant [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P$, and
(3) $\Theta, \overrightarrow{\beta^{\pm}}, \overrightarrow{\gamma^{\pm}} \vdash Q \geqslant [\overrightarrow{\gamma^{\pm}}/\overrightarrow{\alpha^{\pm}}]P$.

$$\begin{aligned}
\mathbf{fv}\, Q &\subseteq \mathbf{fv}\, [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P \cap \mathbf{fv}\, [\overrightarrow{\gamma^{\pm}}/\overrightarrow{\alpha^{\pm}}]P && \text{since by lemma 15, } \mathbf{fv}\, Q \subseteq \mathbf{fv}\, [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P \\
&&& \text{and } \mathbf{fv}\, Q \subseteq \mathbf{fv}\, [\overrightarrow{\gamma^{\pm}}/\overrightarrow{\alpha^{\pm}}]P \\
&\subseteq ((\mathbf{fv}\, P \setminus \overrightarrow{\alpha^{\pm}}) \cup \overrightarrow{\beta^{\pm}}) \cap ((\mathbf{fv}\, P \setminus \overrightarrow{\alpha^{\pm}}) \cup \overrightarrow{\gamma^{\pm}}) \\
&= (\mathbf{fv}\, P \setminus \overrightarrow{\alpha^{\pm}}) \cap (\mathbf{fv}\, P \setminus \overrightarrow{\alpha^{\pm}}) && \text{since } \overrightarrow{\beta^{\pm}} \text{ and } \overrightarrow{\gamma^{\pm}} \text{ are fresh} \\
&= \mathbf{fv}\, P \setminus \overrightarrow{\alpha^{\pm}} \\
&\subseteq T \setminus \overrightarrow{\alpha^{\pm}} && \text{since } P \text{ is well-formed in } T \\
&\subseteq \Theta
\end{aligned}$$

This way, by lemma 2, $\Theta \vdash Q$.

Let us apply $\overrightarrow{\alpha^{\pm}}/\overrightarrow{\beta^{\pm}}$—the inverse of the substitution $\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}$ to both sides of $\Theta, \overrightarrow{\beta^{\pm}}, \overrightarrow{\gamma^{\pm}} \vdash Q \geqslant [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P$ and by lemma 21 (since $\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}$ can be specified as $\Theta, \overrightarrow{\beta^{\pm}}, \overrightarrow{\gamma^{\pm}} \vdash \overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}} : \Theta, \overrightarrow{\alpha^{\pm}}, \overrightarrow{\gamma^{\pm}}$ by lemma 12) obtain $\Theta, \overrightarrow{\alpha^{\pm}}, \overrightarrow{\gamma^{\pm}} \vdash [\overrightarrow{\alpha^{\pm}}/\overrightarrow{\beta^{\pm}}]Q \geqslant P$. Notice that $\Theta \vdash Q$ implies that $\mathbf{fv}\, Q \cap \overrightarrow{\beta^{\pm}} = \emptyset$, then by corollary 3, $[\overrightarrow{\alpha^{\pm}}/\overrightarrow{\beta^{\pm}}]Q = Q$, and thus $\Theta, \overrightarrow{\alpha^{\pm}}, \overrightarrow{\gamma^{\pm}} \vdash Q \geqslant P$. By context strengthening, $\Theta, \overrightarrow{\alpha^{\pm}} \vdash Q \geqslant P$. □

Lemma 75 (Completeness and Initiality of Upgrade). *The upgrade returns the least T-supertype of P well-formed in Θ. Assuming P is well-formed in $T = \Theta, \overrightarrow{\alpha^{\pm}}$,*
For any Q' such that

(1) $\Theta \vdash Q'$ *and*
(2) $T \vdash Q' \geqslant P$,

the result of the upgrade algorithm Q exists ($\mathbf{upgrade}\, T \vdash P\, \mathbf{to}\, \Theta = Q$) *and satisfies* $\Theta \vdash Q' \geqslant Q$.

Proof. Let us consider fresh (not intersecting with T) $\overrightarrow{\beta^{\pm}}$ and $\overrightarrow{\gamma^{\pm}}$.

If we apply substitution $\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}$ to both sides of $\Theta, \overrightarrow{\alpha^{\pm}} \vdash Q' \geqslant P$, we have $\Theta, \overrightarrow{\beta^{\pm}} \vdash [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]Q' \geqslant [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P$, which by corollary 3, since $\overrightarrow{\alpha^{\pm}}$ is disjoint from $\mathbf{fv}\, (Q')$ (because $\Theta \vdash Q'$), simplifies to $\Theta, \overrightarrow{\beta^{\pm}} \vdash Q' \geqslant [\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P$.

Analogously, if we apply substitution $\overrightarrow{\gamma^{\pm}}/\overrightarrow{\alpha^{\pm}}$ to both sides of $\Theta, \overrightarrow{\alpha^{\pm}} \vdash Q' \geqslant P$, we have $\Theta, \overrightarrow{\gamma^{\pm}} \vdash Q' \geqslant [\overrightarrow{\gamma^{\pm}}/\overrightarrow{\alpha^{\pm}}]P$.

This way, Q' is a common supertype of $[\overrightarrow{\beta^{\pm}}/\overrightarrow{\alpha^{\pm}}]P$ and $[\overrightarrow{\gamma^{\pm}}/\overrightarrow{\alpha^{\pm}}]P$ in context $\Theta, \overrightarrow{\beta^{\pm}}, \overrightarrow{\gamma^{\pm}}$. It means that we can apply the completeness of the least upper bound (lemma 73):

(1) there exists Q s.t. $T \vDash [\overrightarrow{\beta^\pm}/\overrightarrow{\alpha^\pm}]P \vee [\overrightarrow{\gamma^\pm}/\overrightarrow{\alpha^\pm}]P = Q$
(2) $T \vdash Q' \geqslant Q$.

The former means that the upgrade algorithm terminates and returns Q. The latter means that since both Q' and Q are well-formed in Θ and T, by lemma 18, $\Theta \vdash Q' \geqslant Q$. □

9.10 Constraint Satisfaction

Lemma 76 (Any constraint is satisfiable). *Suppose that* $\Sigma \vdash C$ *and* Υ *is a set such that* $\mathbf{dom}\,(C) \subseteq \Upsilon \subseteq \mathbf{dom}\,(\Sigma)$. *Then there exists* $\widehat{\sigma}$ *such that* $\Sigma \vdash \widehat{\sigma} : \Upsilon$ *and* $\Sigma \vdash \widehat{\sigma} : C$.

Proof. Let us define $\widehat{\sigma}$ on $\mathbf{dom}\,(C)$ in the following way:

$$[\widehat{\sigma}]\widehat{\alpha}^\pm = \begin{cases} P & \text{if } (\widehat{\alpha}^\pm :\simeq P) \in C \\ P & \text{if } (\widehat{\alpha}^\pm :\geqslant P) \in C \\ N & \text{if } (\widehat{\alpha}^\pm :\simeq N) \in C \\ \exists \beta^-.\, {\downarrow}\beta^- & \text{if } \widehat{\alpha}^\pm = \widehat{\alpha}^+ \in \Upsilon \setminus \mathbf{dom}\,(C) \\ \forall \beta^+.\, {\uparrow}\beta^+ & \text{if } \widehat{\alpha}^\pm = \widehat{\alpha}^- \in \Upsilon \setminus \mathbf{dom}\,(C) \end{cases}$$

Then $\Sigma \vdash \widehat{\sigma} : C$ follows immediately from the reflexivity of equivalence and subtyping (lemma 20) and the corresponding rules $(:\simeq_+^{\text{SAT}})$, $(:\simeq_-^{\text{SAT}})$, and $(:\geqslant_+^{\text{SAT}})$. □

Lemma 77 (Constraint Entry Satisfaction is Stable under Equivalence).

− *If* $T \vdash N_1 : e$ *and* $T \vdash N_1 \simeq^{\leqslant} N_2$ *then* $T \vdash N_2 : e$.
+ *If* $T \vdash P_1 : e$ *and* $T \vdash P_1 \simeq^{\leqslant} P_2$ *then* $T \vdash P_2 : e$.

Proof. − Then e has form $(\widehat{\alpha}^- :\simeq M)$, and by inversion, $T \vdash N_1 \simeq^{\leqslant} M$. Then by transitivity, $T \vdash N_2 \simeq^{\leqslant} M$, meaning $T \vdash N_2 : e$.

+ Let us consider what form e has.
  Case 1. $e = (\widehat{\alpha}^+ :\simeq Q)$. Then $T \vdash P_1 \simeq^{\leqslant} Q$, and hence, $T \vdash P_2 \simeq^{\leqslant} Q$ by transitivity. Then $T \vdash P_2 : e$.
  Case 2. $e = (\widehat{\alpha}^+ :\geqslant Q)$. Then $T \vdash P_1 \geqslant Q$, and hence, $T \vdash P_2 \geqslant Q$ by transitivity. Then $T \vdash P_2 : e$.

□

Corollary 28 (Constraint Satisfaction is stable under Equivalence).
If $\Sigma \vdash \widehat{\sigma}_1 : C$ *and* $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{dom}\,(C)$ *then* $\Sigma \vdash \widehat{\sigma}_2 : C$;
if $\Sigma \vdash \widehat{\sigma}_1 : UC$ *and* $\Sigma \vdash \widehat{\sigma}_1 \simeq^{\leqslant} \widehat{\sigma}_2 : \mathbf{dom}\,(C)$ *then* $\Sigma \vdash \widehat{\sigma}_2 : UC$.

Corollary 29 (Normalization preserves Constraint Satisfaction).
If $\Sigma \vdash \widehat{\sigma} : C$ *then* $\Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}) : C$;
if $\Sigma \vdash \widehat{\sigma} : UC$ *then* $\Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}) : UC$.

9.11 Positive Subtyping

Observation 10 (Positive Subtyping is Deterministic). *For fixed* T, Σ, P, *and* Q, *if* $T; \Sigma \vDash P \geqslant Q \dashv C$ *and* $T; \Sigma \vDash P \geqslant Q \dashv C'$ *then* $C = C'$.

Proof. We prove it by induction on $T; \Sigma \vDash P \geqslant Q \dashv C$. First, it is easy to see that the rule applied to infer $T; \Sigma \vDash P \geqslant Q \dashv C$ uniquely depends on the input, and those, it is the same rule that is inferring $T; \Sigma \vDash P \geqslant Q \dashv C'$. Second, the premises of each rule are deterministic on the input: unification is deterministic by observation 4, upgrade is deterministic by observation 8, the choice of the fresh algorithmic variables is deterministic by convention, as discussed in section 2.2, positive subtyping by the induction hypothesis. □

Lemma 78 (Soundness of the Positive Subtyping). *If $T \vdash^{\supseteq} \Sigma$, $T \vdash Q$, T ; $\mathbf{dom}(\Sigma) \vdash P$, and $T; \Sigma \models P \geqslant Q \dashv C$, then $\Sigma \vdash C : \mathbf{fav} P$ and for any normalized $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : C$, $T \vdash [\widehat{\sigma}] P \geqslant Q$.*

Proof. We prove it by induction on $T; \Sigma \models P \geqslant Q \dashv C$. Let us consider the last rule to infer this judgment.

Case 1. ($\text{UVar}^{\geqslant}$) then $T; \Sigma \models P \geqslant Q \dashv C$ has shape $T; \Sigma \models \widehat{\alpha}^+ \geqslant P' \dashv (\widehat{\alpha}^+ :\geqslant Q')$ where $\widehat{\alpha}^+\{\Theta\} \in \Sigma$ and **upgrade** $T \vdash P'$ **to** $\Theta = Q'$.
Notice that $\widehat{\alpha}^+\{\Theta\} \in \Sigma$ and $T \vdash^{\supseteq} \Sigma$ implies $T = \Theta, \overrightarrow{\alpha^{\pm}}$ for some $\overrightarrow{\alpha^{\pm}}$, hence, the soundness of upgrade (lemma 95) is applicable:
(1) $\Theta \vdash Q'$ and
(2) $T \vdash Q' \geqslant P$.
Since $\widehat{\alpha}^+\{\Theta\} \in \Sigma$ and $\Theta \vdash Q'$, it is clear that $\Sigma \vdash (\widehat{\alpha}^+ :\geqslant Q') : \widehat{\alpha}^+$.
It is left to show that $T \vdash [\widehat{\sigma}]\widehat{\alpha}^+ \geqslant P'$ for any normalized $\widehat{\sigma}$ s.t. $\Sigma \vdash \widehat{\sigma} : (\widehat{\alpha}^+ :\geqslant Q')$. The latter means that $\Sigma(\widehat{\alpha}^+) \vdash [\widehat{\sigma}]\widehat{\alpha}^+ \geqslant Q'$, i.e. $\Theta \vdash [\widehat{\sigma}]\widehat{\alpha}^+ \geqslant Q'$. By weakening the context to T and combining this judgment transitively with $T \vdash Q' \geqslant P$, we have $T \vdash [\widehat{\sigma}]\widehat{\alpha}^+ \geqslant P$, as required.

Case 2. ($\text{Var}^{\geqslant}_{+}$) then $T; \Sigma \models P \geqslant Q \dashv C$ has shape $T; \Sigma \models \alpha^+ \geqslant \alpha^+ \dashv \cdot$. Then $\mathbf{fav}\alpha^+ = \emptyset$, and $C = \cdot$ satisfies $\Sigma \vdash C : \cdot$. Since $\mathbf{fav}\alpha^+ = \emptyset$, application of any substitution $\widehat{\sigma}$ does not change α^+, i.e. $[\widehat{\sigma}]\alpha^+ = \alpha^+$. Therefore, $T \vdash [\widehat{\sigma}]\alpha^+ \geqslant \alpha^+$ holds by ($\text{Var}^{\leqslant}_{-}$).

Case 3. ($\downarrow^{\geqslant}$) then $T; \Sigma \models P \geqslant Q \dashv C$ has shape $T; \Sigma \models \downarrow N \geqslant \downarrow M \dashv C$.
Then the next step of the algorithm is the unification of $\mathbf{nf}(N)$ and $\mathbf{nf}(M)$, and it returns the resulting unification constraint $UC = C$ as the result. By the soundness of unification (lemma 64), $\Sigma \vdash C : \mathbf{fav}(N)$ and for any normalized $\widehat{\sigma}$, $\Sigma \vdash \widehat{\sigma} : C$ implies $[\widehat{\sigma}]\mathbf{nf}(N) = \mathbf{nf}(M)$, then we rewrite the left-hand side by lemma 41: $\mathbf{nf}([\widehat{\sigma}]N) = \mathbf{nf}(M)$ and apply lemma 46: $T \vdash [\widehat{\sigma}]N \simeq^{\leqslant} M$, then by ($\uparrow^{\leqslant}$), $T \vdash \downarrow[\widehat{\sigma}]N \geqslant \downarrow M$.

Case 4. ($\exists^{\geqslant}$) then $T; \Sigma \models P \geqslant Q \dashv C$ has shape $T; \Sigma \models \exists\overrightarrow{\alpha^-}. P' \geqslant \exists\overrightarrow{\beta^-}. Q' \dashv C$ s.t. either $\overrightarrow{\alpha^-}$ or $\overrightarrow{\beta^-}$ is not empty.
Then the algorithm creates fresh unification variables $\overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\}$, substitutes the old $\overrightarrow{\alpha^-}$ with them in P', and makes the recursive call: $T, \overrightarrow{\beta^-}; \Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\} \models [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]P' \geqslant Q' \dashv C'$, returning as the result $C = C' \setminus \overrightarrow{\widehat{\alpha}^-}$.
Let us take an arbitrary normalized $\widehat{\sigma}$ s.t. $\Sigma \vdash \widehat{\sigma} : C' \setminus \overrightarrow{\widehat{\alpha}^-}$. We wish to show $T \vdash [\widehat{\sigma}]P \geqslant Q$, i.e. $T \vdash \exists\overrightarrow{\alpha^-}. [\widehat{\sigma}]P' \geqslant \exists\overrightarrow{\beta^-}. Q'$. To do that, we apply ($\exists^{\geqslant}$), and what is left to show is $T, \overrightarrow{\beta^-} \vdash [\overrightarrow{N}/\overrightarrow{\alpha^-}][\widehat{\sigma}]P' \geqslant Q'$ for some $\overrightarrow{N}$. If we construct a normalized $\widehat{\sigma}'$ such that $\Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\} \vdash \widehat{\sigma}' : C'$ and for some $\overrightarrow{N}$, $[\overrightarrow{N}/\overrightarrow{\alpha^-}][\widehat{\sigma}]P' = [\widehat{\sigma}'][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]P'$, we can apply the induction hypothesis to $T, \overrightarrow{\beta^-}; \Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\} \models [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]P \geqslant Q \dashv C'$ and infer the required subtyping.
Let us construct such $\widehat{\sigma}'$ by extending $\widehat{\sigma}$ with $\overrightarrow{\widehat{\alpha}^-}$ mapped to the corresponding types in C':

$$[\widehat{\sigma}']\widehat{\beta}^{\pm} = \begin{cases} [\widehat{\sigma}]\widehat{\beta}^{\pm} & \text{if } \widehat{\beta}^{\pm} \in \mathbf{dom}(C') \setminus \overrightarrow{\widehat{\alpha}} \\ \mathbf{nf}(N) & \text{if } \widehat{\beta}^{\pm} \in \overrightarrow{\widehat{\alpha}^-} \text{ and } (\widehat{\beta}^{\pm} :\simeq N) \in SC' \end{cases}$$

It is easy to see that $\widehat{\sigma}'$ is normalized: it inherits this property from $\widehat{\sigma}$. Let us show that $\Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\} \vdash \widehat{\sigma}' : C'$. Let us take an arbitrary entry e from C' restricting a variable $\widehat{\beta}^{\pm}$. Suppose $\widehat{\beta}^{\pm} \in \mathbf{dom}(C') \setminus \overrightarrow{\widehat{\alpha}^-}$. Then $(\Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\})(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}']\widehat{\beta}^{\pm} : e$ is rewritten as $\Sigma(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}]\widehat{\beta}^{\pm} : e$, which holds since $\Sigma \vdash \widehat{\sigma} : C'$. Suppose $\widehat{\beta}^{\pm} = \widehat{\alpha}_i^- \in \overrightarrow{\widehat{\alpha}^-}$. Then $e = (\widehat{\alpha}_i^- :\simeq N)$ for some N, $[\widehat{\sigma}']\widehat{\alpha}_i^- = \mathbf{nf}(N)$ by the definition, and $T, \overrightarrow{\beta^-} \vdash \mathbf{nf}(N) : (\widehat{\alpha}_i^- :\simeq N)$ by ($:\simeq^{\text{SAT}}_{-}$), since $T \vdash \mathbf{nf}(N) \simeq^{\leqslant} N$ by lemma 46.

Finally, let us show that $[\overrightarrow{N}/\overrightarrow{\alpha^-}][\widehat{\sigma}]P' = [\widehat{\sigma}'][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]P'$. For N_i, we take the *normalized* type restricting $\widehat{\alpha}_i^-$ in C'. Let us take an arbitrary variable from P.

(1) If this variable is a unification variable $\widehat{\beta}^\pm$, then $[\overrightarrow{N}/\overrightarrow{\alpha^-}][\widehat{\sigma}]\widehat{\beta}^\pm = [\widehat{\sigma}]\widehat{\beta}^\pm$, since $\Sigma \vdash \widehat{\sigma} : C' \setminus \overrightarrow{\widehat{\alpha}^-}$ and $\mathbf{dom}\,(\Sigma) \cap \overrightarrow{\alpha^-} = \emptyset$.
Notice that $\widehat{\beta}^\pm \in \mathbf{dom}\,(\Sigma)$, which is disjoint from $\overrightarrow{\widehat{\alpha}^-}$, that is $\widehat{\beta}^\pm \in \mathbf{dom}\,(C') \setminus \overrightarrow{\widehat{\alpha}^-}$. This way, $[\widehat{\sigma}'][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\widehat{\beta}^\pm = [\widehat{\sigma}']\widehat{\beta}^\pm = [\widehat{\sigma}]\widehat{\beta}^\pm$ by the definition of $\widehat{\sigma}'$,

(2) If this variable is a regular variable $\beta^\pm \notin \overrightarrow{\alpha^-}$, then $[\overrightarrow{N}/\overrightarrow{\alpha^-}][\widehat{\sigma}]\beta^\pm = \beta^\pm$ and $[\widehat{\sigma}'][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\beta^\pm = \beta^\pm$.

(3) If this variable is a regular variable $\alpha_i^- \in \overrightarrow{\alpha^-}$, then $[\overrightarrow{N}/\overrightarrow{\alpha^-}][\widehat{\sigma}]\alpha_i^- = N_i = \mathbf{nf}\,(N_i)$ (the latter equality holds since N_i is normalized) and $[\widehat{\sigma}'][\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]\alpha_i^- = [\widehat{\sigma}']\widehat{\alpha}_i^- = \mathbf{nf}\,(N_i)$.

□

Lemma 79 (Completeness of the Positive Subtyping). *Suppose that* $T \vdash^{\supseteq} \Sigma$, $T \vdash Q$ *and* $T\,; \mathbf{dom}\,(\Sigma) \vdash P$. *Then for any* $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P)$ *such that* $T \vdash [\widehat{\sigma}]P \geqslant Q$, *there exists* $T;\, \Sigma \models P \geqslant Q \dashv C$ *and moreover,* $\Sigma \vdash \widehat{\sigma} : C$.

Proof. Let us prove this lemma by induction on $T \vdash [\widehat{\sigma}]P \geqslant Q$. Let us consider the last rule used in the derivation, but first, consider the base case for the substitution $[\widehat{\sigma}]P$:

Case 1. $P = \exists\overrightarrow{\beta^-}.\,\widehat{\alpha}^+$ (for potentially empty $\overrightarrow{\beta^-}$)
Then by assumption, $T \vdash \exists\overrightarrow{\beta^-}.\,[\widehat{\sigma}]\widehat{\alpha}^+ \geqslant Q$ (where $\overrightarrow{\beta^-} \cap \mathbf{fv}\,[\widehat{\sigma}]\widehat{\alpha}^+ = \emptyset$). Let us decompose Q as $Q = \exists\overrightarrow{\gamma^-}.\,Q_0$, where Q_0 does not start with $\exists$.
By inversion, $T\,; \mathbf{dom}\,(\Sigma) \vdash \exists\overrightarrow{\beta^-}.\,\widehat{\alpha}^+$ implies $\widehat{\alpha}^+\{\Theta\} \in \Sigma$ for some $\Theta \subseteq T$.
By lemma 16 applied twice, $T \vdash \exists\overrightarrow{\beta^-}.\,[\widehat{\sigma}]\widehat{\alpha}^+ \geqslant \exists\overrightarrow{\gamma^-}.\,Q_0$ implies $T, \overrightarrow{\gamma^-} \vdash [\overrightarrow{N}/\overrightarrow{\beta^-}][\widehat{\sigma}]\widehat{\alpha}^+ \geqslant Q_0$ for some $\overrightarrow{N}$, and since $\overrightarrow{\beta^-} \cap \mathbf{fv}\,([\widehat{\sigma}]\widehat{\alpha}^+) \subseteq \overrightarrow{\beta^-} \cap \Sigma(\widehat{\alpha}^+) \subseteq \overrightarrow{\beta^-} \cap T = \emptyset$, $[\overrightarrow{N}/\overrightarrow{\beta^-}][\widehat{\sigma}]\widehat{\alpha}^+ = [\widehat{\sigma}]\widehat{\alpha}^+$, that is $T, \overrightarrow{\gamma^-} \vdash [\widehat{\sigma}]\widehat{\alpha}^+ \geqslant Q_0$.
When algorithm tires to infer the subtyping $T;\, \Sigma \models \exists\overrightarrow{\beta^-}.\,\widehat{\alpha}^+ \geqslant \exists\overrightarrow{\gamma^-}.\,Q_0 \dashv C$, it applies $(\exists^{\geqslant})$, which reduces the problem to $T, \overrightarrow{\gamma^-};\, \Sigma, \overrightarrow{\widehat{\beta}^-}\{T, \overrightarrow{\gamma^-}\} \models [\overrightarrow{\widehat{\beta}^-}/\overrightarrow{\beta^-}]\widehat{\alpha}^+ \geqslant Q_0 \dashv C$, which is equivalent to $T, \overrightarrow{\gamma^-};\, \Sigma, \overrightarrow{\widehat{\beta}^-}\{T, \overrightarrow{\gamma^-}\} \models \widehat{\alpha}^+ \geqslant Q_0 \dashv C$.
Next, the algorithm tries to apply $(\text{UVar}^{\geqslant})$ and the resulting restriction is $C = (\widehat{\alpha}^+ :\geqslant Q_0')$ where **upgrade** $T, \overrightarrow{\gamma^-} \vdash Q_0$ **to** $\Theta = Q_0'$.
Why does the upgrade procedure terminate? Because $[\widehat{\sigma}]\widehat{\alpha}^+$ satisfies the pre-conditions of the completeness of the upgrade (lemma 75):

(1) $\Theta \vdash [\widehat{\sigma}]\widehat{\alpha}^+$ because $\Sigma \vdash \widehat{\sigma} : \widehat{\alpha}^+$ and $\widehat{\alpha}^+\{\Theta\} \in \Sigma$,
(2) $T, \overrightarrow{\gamma^-} \vdash [\widehat{\sigma}]\widehat{\alpha}^+ \geqslant Q_0$ as noted above

Moreover, the completeness of upgrade also says that Q_0' is *the least* supertype of Q_0 among types well-formed in Θ, that is $\Theta \vdash [\widehat{\sigma}]\widehat{\alpha}^+ \geqslant Q_0'$, which means $\Sigma \vdash \widehat{\sigma} : (\widehat{\alpha}^+ :\geqslant Q_0')$, that is $\Sigma \vdash \widehat{\sigma} : C$.

Case 2. $T \vdash [\widehat{\sigma}]P \geqslant Q$ is derived by $(\text{Var}_+^{\geqslant})$
Then $P = [\widehat{\sigma}]P = \alpha^+ = Q$, where the first equality holds because P is not a unification variable: it has been covered by case 1; and the second equality hold because $(\text{Var}_+^{\geqslant})$ was applied.
The algorithm applies $(\text{Var}_+^{\geqslant})$ and infers $C = \cdot$, i.e. $T;\, \Sigma \models \alpha^+ \geqslant \alpha^+ \dashv \cdot$. Then $\Sigma \vdash \widehat{\sigma} : \cdot$ holds trivially.

Case 3. $T \vdash [\widehat{\sigma}]P \geqslant Q$ is derived by $(\downarrow^{\geqslant})$,

Then $P = {\downarrow}N$, since the substitution $[\widehat{\sigma}]P$ must preserve the top-level constructor of $P \neq \widehat{\alpha}^+$ (the case $P = \widehat{\alpha}^+$ has been covered by case 1), and $Q = {\downarrow}M$, and by inversion, $T \vdash [\widehat{\sigma}]N \simeq^{\leqslant} M$.
Since both types start with $\downarrow$, the algorithm tries to apply $(\downarrow^{\geqslant})$: $T;\, \Sigma \vDash {\downarrow}N \geqslant {\downarrow}M \dashv C$. The premise of this rule is the unification of $\mathbf{nf}\,(N)$ and $\mathbf{nf}\,(M)$: $T; \Sigma \vDash \mathbf{nf}\,(N) \overset{u}{\simeq} \mathbf{nf}\,(M) \dashv UC$. And the algorithm returns it as a subtyping constraint $C = UC$.
To demonstrate that the unification terminates ant $\widehat{\sigma}$ satisfies the resulting constraints, we apply the completeness of the unification algorithm (lemma 65). In order to do that, we need to provide a substitution unifying $\mathbf{nf}\,(N)$ and $\mathbf{nf}\,(M)$. Let us show that $\mathbf{nf}\,(\widehat{\sigma})$ is such a substitution.

- $\mathbf{nf}\,(N)$ and $\mathbf{nf}\,(M)$ are normalized
- $T\,;\mathbf{dom}\,(\Sigma) \vdash \mathbf{nf}\,(N)$ because $T\,;\mathbf{dom}\,(\Sigma) \vdash N$ (corollary 24)
- $T \vdash \mathbf{nf}\,(M)$ because $T \vdash M$ (corollary 14)
- $\Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}) : \mathbf{fav}(P)$ because $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P)$ (corollary 25)
- $T \vdash [\widehat{\sigma}]N \simeq^{\leqslant} M \Rightarrow [\widehat{\sigma}]N \simeq^{D} M$ by lemma 32

  $\Rightarrow \mathbf{nf}\,([\widehat{\sigma}]N) = \mathbf{nf}\,(M)$ by lemma 42

  $\Rightarrow [\mathbf{nf}\,(\widehat{\sigma})]\mathbf{nf}\,(N) = \mathbf{nf}\,(M)$ by lemma 41

By the completeness of the unification, $T; \Sigma \vDash N \overset{u}{\simeq} M \dashv UC$ exists, and $\Sigma \vdash \mathbf{nf}\,(\widehat{\sigma}) : UC$, and by corollary 28, $\Sigma \vdash \widehat{\sigma} : UC$.

Case 4. $T \vdash [\widehat{\sigma}]P \geqslant Q$ is derived by $(\exists^{\geqslant})$.
We should only consider the case when the substitution $[\widehat{\sigma}]P$ results in the existential type $\exists\overrightarrow{\alpha^-}.\, P''$ (for $P'' \neq \exists \ldots$) by congruence, i.e. $P = \exists\overrightarrow{\alpha^-}.\, P'$ (for $P' \neq \exists \ldots$) and $[\widehat{\sigma}]P' = P''$. This is because the case when $P = \exists\overrightarrow{\beta^-}.\, \widehat{\alpha}^+$ has been covered (case 1), and thus, the substitution $\widehat{\sigma}$ must preserve all the outer quantifiers of P and does not generate any new ones.
This way, $P = \exists\overrightarrow{\alpha^-}.\, P'$, $[\widehat{\sigma}]P = \exists\overrightarrow{\alpha^-}.\, [\widehat{\sigma}]P'$ (assuming $\overrightarrow{\alpha^-}$ does not intersect with the range of $\widehat{\sigma}$) and $Q = \exists\overrightarrow{\beta^-}.\, Q'$, where either $\overrightarrow{\alpha^-}$ or $\overrightarrow{\beta^-}$ is not empty.
By inversion, $T \vdash [\sigma][\widehat{\sigma}]P' \geqslant Q'$ for some $T, \overrightarrow{\beta^-} \vdash \sigma : \overrightarrow{\alpha^-}$. Since σ and $\widehat{\sigma}$ have disjoint domains, and the range of one does not intersect with the domain of the other, they commute, i.e. $T, \overrightarrow{\beta^-} \vdash [\widehat{\sigma}][\sigma]P' \geqslant Q'$ (notice that the tree inferring this judgement is a proper subtree of the tree inferring $T \vdash [\widehat{\sigma}]P \geqslant Q$).
At the next step, the algorithm creates fresh (disjoint with $\mathbf{fav}P'$) unification variables $\overrightarrow{\widehat{\alpha}^-}$, replaces $\overrightarrow{\alpha^-}$ with them in P', and makes the recursive call: $T, \overrightarrow{\beta^-};\, \Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\} \vDash P_0 \geqslant Q' \dashv C_1$, (where $P_0 = [\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]P'$), returning $C_1 \setminus \overrightarrow{\widehat{\alpha}^-}$ as the result.
To show that the recursive call terminates and that $\Sigma \vdash \widehat{\sigma} : C_1 \setminus \overrightarrow{\widehat{\alpha}^-}$, it suffices to build $\Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\} \vdash \widehat{\sigma}_0 : \mathbf{fav}(P_0)$—an extension of $\widehat{\sigma}$ with $\overrightarrow{\widehat{\alpha}^-} \cap \mathbf{fav}(P_0)$ such that $T, \overrightarrow{\beta^-} \vdash [\widehat{\sigma}_0]P_0 \geqslant Q$. Then by the induction hypothesis, $\Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\} \vdash \widehat{\sigma}_0 : C_1$, and hence, $\Sigma \vdash \widehat{\sigma} : C_1 \setminus \overrightarrow{\widehat{\alpha}^-}$, as required.
Let us construct such a substitution $\widehat{\sigma}_0$:

$$[\widehat{\sigma}_0]\widehat{\beta}^{\pm} = \begin{cases} [\sigma]\alpha_i^- & \text{if } \widehat{\beta}^{\pm} = \widehat{\alpha}_i^- \in \overrightarrow{\widehat{\alpha}^-} \cap \mathbf{fav}(P_0) \\ [\widehat{\sigma}]\widehat{\beta}^{\pm} & \text{if } \widehat{\beta}^{\pm} \in \mathbf{fav}(P') \end{cases}$$

It is easy to see $\Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\} \vdash \widehat{\sigma}_0 : \mathbf{fav}(P_0)$: $\mathbf{fav}(P_0) = \mathbf{fav}([\overrightarrow{\widehat{\alpha}^-}/\overrightarrow{\alpha^-}]P') = \overrightarrow{\widehat{\alpha}^-} \cap \mathbf{fav}(P_0) \cup \mathbf{fav}(P')$. Then

(1) for $\widehat{\alpha}_i^- \in \overrightarrow{\widehat{\alpha}^-} \cap \mathbf{fav}(P_0)$, $(\Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\})(\widehat{\alpha}_i^-) \vdash [\widehat{\sigma}_0]\widehat{\alpha}_i^-$, i.e. $T, \overrightarrow{\beta^-} \vdash [\sigma]\alpha_i^-$ holds since $T, \overrightarrow{\beta^-} \vdash \sigma : \overrightarrow{\alpha^-}$,

(2) for $\widehat{\beta}^\pm \in \mathbf{fav}(P') \subseteq \mathbf{dom}(\Sigma)$, $(\Sigma, \overrightarrow{\widehat{\alpha}^-}\{T, \overrightarrow{\beta^-}\})(\widehat{\beta}^\pm) \vdash [\widehat{\sigma}_0]\widehat{\beta}^\pm$, i.e. $\Sigma(\widehat{\beta}^\pm) \vdash [\widehat{\sigma}]\widehat{\beta}^\pm$ holds since $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P)$ and $\widehat{\beta}^\pm \in \mathbf{fav}(P') = \mathbf{fav}(P)$.

Now, let us show that $T, \overrightarrow{\beta^-} \vdash [\widehat{\sigma}_0]P_0 \geqslant Q$. To do that, we notice that $[\widehat{\sigma}_0]P_0 = [\widehat{\sigma}][\sigma][\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}]P_0$: let us consider an arbitrary variable appearing freely in P_0:

(1) if this variable is a algorithmic variable $\widehat{\alpha}_i^- \in \overrightarrow{\widehat{\alpha}^-}$, then $[\widehat{\sigma}_0]\widehat{\alpha}_i^- = [\sigma]\alpha_i^-$ and $[\widehat{\sigma}][\sigma][\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}]\widehat{\alpha}_i^- = [\widehat{\sigma}][\sigma]\alpha_i^- = [\sigma]\alpha_i^-$,

(2) if this variable is a algorithmic variable $\widehat{\beta}^\pm \in \mathbf{fav}(P_0) \setminus \overrightarrow{\widehat{\alpha}^-} = \mathbf{fav}(P')$, then $[\widehat{\sigma}_0]\widehat{\beta}^\pm = [\widehat{\sigma}]\widehat{\beta}^\pm$ and $[\widehat{\sigma}][\sigma][\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}]\widehat{\beta}^\pm = [\widehat{\sigma}][\sigma]\widehat{\beta}^\pm = [\widehat{\sigma}]\widehat{\beta}^\pm$,

(3) if this variable is a regular variable from $\mathbf{fv}(P_0)$, both substitutions do not change it: $\widehat{\sigma}_0$, $\widehat{\sigma}$ and $\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}$ act on algorithmic variables, and σ is defined on $\overrightarrow{\alpha^-}$, however, $\overrightarrow{\alpha^-} \cap \mathbf{fv}(P_0) = \emptyset$.

This way, $[\widehat{\sigma}_0]P_0 = [\widehat{\sigma}][\sigma][\overrightarrow{\alpha^-}/\overrightarrow{\widehat{\alpha}^-}]P_0 = [\widehat{\sigma}][\sigma]P'$, and thus, $T, \overrightarrow{\beta^-} \vdash [\widehat{\sigma}_0]P_0 \geqslant Q'$.

□

9.12 Subtyping Constraint Merge

Observation 11 (Constraint Entry Merge is Deterministic). *For fixed T, e_1, e_2, if $T \vdash e_1 \,\&\, e_2 = e$ and $T \vdash e_1 \,\&\, e_2 = e'$ then $e = e'$.*

Proof. First, notice that the shape of e_1 and e_2 uniquely determines the rule applied to infer $T \vdash e_1 \,\&\, e_2 = e$, which is consequently, the same rule used to infer $T \vdash e_1 \,\&\, e_2 = e'$. Second, notice that the premises of each rule are deterministic on the input: the positive subtyping is deterministic by observation 10, and the least upper bound is deterministic by observation 7. □

Observation 12 (Subtyping Constraint Merge is Deterministic). *Suppose that $\Sigma \vdash C_1$ and $\Sigma \vdash C_2$ If $\Sigma \vdash C_1 \,\&\, C_2 = C$ and $\Sigma \vdash C_1 \,\&\, C_2 = C'$ are defined then $C = C'$.*

Proof. The proof is analogous to the proof of observation 3 but uses observation 11 to show that the merge of the matching constraint entries is fixed. □

Lemma 80 (Soundness of Constraint Entry Merge). *For a fixed context T, suppose that $T \vdash e_1$ and $T \vdash e_2$. If $T \vdash e_1 \,\&\, e_2 = e$ is defined then*

(1) $T \vdash e$

(2) For any $T \vdash P$, $T \vdash P : e$ implies $T \vdash P : e_1$ and $T \vdash P : e_2$

Proof. Let us consider the rule forming $T \vdash e_1 \,\&\, e_2 = e$.

Case 1. $(\simeq \&^+ \simeq)$, i.e. $T \vdash e_1 \,\&\, e_2 = e$ has form $T \vdash (\widehat{\alpha}^+ :\simeq Q) \,\&\, (\widehat{\alpha}^+ :\simeq Q') = (\widehat{\alpha}^+ :\simeq Q)$ and $\mathbf{nf}(Q) = \mathbf{nf}(Q')$. The latter implies $T \vdash Q \simeq^\leqslant Q'$ by lemma 46. Then

(1) $T \vdash e$, i.e. $T \vdash \widehat{\alpha}^+ :\simeq Q$ holds by assumption;

(2) by inversion, $T \vdash P : (\widehat{\alpha}^+ :\simeq Q)$ means $T \vdash P \simeq^\leqslant Q$, and by transitivity of equivalence (corollary 10), $T \vdash P \simeq^\leqslant Q'$. Thus, $T \vdash P : e_1$ and $T \vdash P : e_2$ hold by $(:\simeq^{\text{SAT}}_+)$.

Case 2. $(\simeq \&^- \simeq)$ the negative case is proved in exactly the same way as the positive one.

Case 3. $(\geqslant \&^+ \geqslant)$ Then e_1 is $\widehat{\alpha}^+ :\geqslant Q_1$, e_2 is $\widehat{\alpha}^+ :\geqslant Q_2$, and $e_1 \,\&\, e_2 = e$ is $\widehat{\alpha}^+ :\geqslant Q$ where Q is the least upper bound of Q_1 and Q_2. Then by lemma 72,

- $T \vdash Q$,
- $T \vdash Q \geqslant Q_1$,
- $T \vdash Q \geqslant Q_2$.

Let us show the required properties.

- $T \vdash e$ holds from $T \vdash Q$,
- Assuming $T \vdash P : e$, by inversion, we have $T \vdash P \geqslant Q$. Combining it transitively with $T \vdash Q \geqslant Q_1$, we have $T \vdash P \geqslant Q_1$. Analogously, $T \vdash P \geqslant Q_2$. Then $T \vdash P : e_1$ and $T \vdash P : e_2$ hold by $(:\geqslant^{\text{SAT}}_{+})$.

Case 4. $(\geqslant \,\&^{+} \simeq)$ Then e_1 is $\widehat{\alpha}^{+} :\geqslant Q_1$, e_2 is $\widehat{\alpha}^{+} :\simeq Q_2$, where $T; \cdot \vDash Q_2 \geqslant Q_1 \dashv \cdot$, and the resulting $e_1 \,\&\, e_2 = e$ is equal to e_2, that is $\widehat{\alpha}^{+} :\simeq Q_2$.

Let us show the required properties.

- By assumption, $T \vdash Q$, and hence $T \vdash e$.
- Since $\mathbf{fav}(Q_2) = \emptyset$, $T; \cdot \vDash Q_2 \geqslant Q_1 \dashv \cdot$ implies $T \vdash Q_2 \geqslant Q_1$ by the soundness of positive subtyping (lemma 78). Then let us take an arbitrary $T \vdash P$ such that $T \vdash P : e$. Since $e_2 = e$, $T \vdash P : e_2$ holds immediately.
  By inversion, $T \vdash P : (\widehat{\alpha}^{+} :\simeq Q_2)$ means $T \vdash P \simeq^{\leqslant} Q_2$, and then by transitivity of subtyping (lemma 22), $T \vdash P \geqslant Q_1$. Then $T \vdash P : e_1$ holds by $(:\geqslant^{\text{SAT}}_{+})$.

Case 5. $(\simeq \,\&^{+} \geqslant)$ Thee proof is analogous to the previous case.

□

Lemma 81 (Soundness of Constraint Merge)**.** *Suppose that $\Sigma \vdash C_1 : \Upsilon_1$ and $\Sigma \vdash C_2 : \Upsilon_2$ and $\Sigma \vdash C_1 \,\&\, C_2 = C$ is defined. Then*

(1) $\Sigma \vdash C : \Upsilon_1 \cup \Upsilon_2$,
(2) for any substitution $\Sigma \vdash \widehat{\sigma} : \Upsilon_1 \cup \Upsilon_2$, $\Sigma \vdash \widehat{\sigma} : C$ implies $\Sigma \vdash \widehat{\sigma} : C_1$ and $\Sigma \vdash \widehat{\sigma} : C_2$.

Proof. By definition, $\Sigma \vdash C_1 \,\&\, C_2 = C$ consists of three parts: entries of C_1 that do not have matching entries of C_2, entries of C_2 that do not have matching entries of C_1, and the merge of matching entries.

Notice that $\widehat{\alpha}^{\pm} \in \Upsilon_1 \setminus \Upsilon_2$ if and only if there is an entry e in C_1 restricting $\widehat{\alpha}^{\pm}$, but there is no such entry in C_2. Therefore, for any $\widehat{\alpha}^{\pm} \in \Upsilon_1 \setminus \Upsilon_2$, there is an entry e in C restricting $\widehat{\alpha}^{\pm}$. Notice that $\Sigma(\widehat{\alpha}^{\pm}) \vdash e$ holds since $\Sigma \vdash C_1 : \Upsilon_1$.

Analogously, for any $\widehat{\beta}^{\pm} \in \Upsilon_2 \setminus \Upsilon_1$, there is an entry e in C restricting $\widehat{\beta}^{\pm}$. Notice that $\Sigma(\widehat{\beta}^{\pm}) \vdash e$ holds since $\Sigma \vdash C_2 : \Upsilon_2$.

Finally, for any $\widehat{\gamma}^{\pm} \in \Upsilon_1 \cap \Upsilon_2$, there is an entry e_1 in C_1 restricting $\widehat{\gamma}^{\pm}$ and an entry e_2 in C_2 restricting $\widehat{\gamma}^{\pm}$. Since $\Sigma \vdash C_1 \,\&\, C_2 = C$ is defined, $\Sigma(\widehat{\gamma}^{\pm}) \vdash e_1 \,\&\, e_2 = e$ restricting $\widehat{\gamma}^{\pm}$ is defined and belongs to C, moreover, $\Sigma(\widehat{\gamma}^{\pm}) \vdash e$ by lemma 80. This way, $\Sigma \vdash C : \Upsilon_1 \cup \Upsilon_2$.

Let us show the second property. We take an arbitrary $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : \Upsilon_1 \cup \Upsilon_2$ and $\Sigma \vdash \widehat{\sigma} : C$. To prove $\Sigma \vdash \widehat{\sigma} : C_1$, we need to show that for any $e_1 \in C_1$, restricting $\widehat{\alpha}^{\pm}$, $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}]\widehat{\alpha}^{\pm} : e_1$ holds.

Let us assume that $\widehat{\alpha}^{\pm} \notin \mathbf{dom}\,(C_2)$. It means that $C \ni e_1$, and then since $\Sigma \vdash \widehat{\sigma} : C$, $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}]\widehat{\alpha}^{\pm} : e_1$.

Otherwise, C_2 contains an entry e_2 restricting $\widehat{\alpha}^{\pm}$, and $C \ni e$ where $\Sigma(\widehat{\alpha}^{\pm}) \vdash e_1 \,\&\, e_2 = e$. Then since $\Sigma \vdash \widehat{\sigma} : C$, $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}]\widehat{\alpha}^{\pm} : e$, and by lemma 80, $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}]\widehat{\alpha}^{\pm} : e_1$.

The proof of $\Sigma \vdash \widehat{\sigma} : C_2$ is symmetric. □

Lemma 82 (Completeness of Constraint Entry Merge)**.** *For a fixed context T, suppose that $T \vdash e_1$ and $T \vdash e_2$ are matching constraint entries.*

- *for a type P such that $T \vdash P : e_1$ and $T \vdash P : e_2$, $T \vdash e_1 \,\&\, e_2 = e$ is defined and $T \vdash P : e$.*
- *for a type N such that $T \vdash N : e_1$ and $T \vdash N : e_2$, $T \vdash e_1 \,\&\, e_2 = e$ is defined and $T \vdash N : e$.*

Proof. Let us consider the shape of e_1 and e_2.

Case 1. e_1 is $\widehat{\alpha}^{+} :\simeq Q_1$ and e_2 is $\widehat{\alpha}^{+} :\simeq Q_2$. The proof repeats the corresponding case of lemma 62

Case 2. e_1 is $\widehat{\alpha}^+ :\simeq Q_1$ and e_2 is $\widehat{\alpha}^+ :\geqslant Q_2$. Then $T \vdash P : e_1$ means $T \vdash P \simeq^{\leqslant} Q_1$, and $T \vdash P : e_2$ means $T \vdash P \geqslant Q_2$. Then by transitivity of subtyping, $T \vdash Q_1 \geqslant Q_2$, which means $T; \cdot \vDash Q_1 \geqslant Q_2 \dashv \cdot$ by lemma 79. This way, ($\simeq \&^+ \geqslant$) applies to infer $T \vdash e_1 \& e_2 = e_1$, and $T \vdash P : e_1$ holds by assumption.

Case 3. e_1 is $\widehat{\alpha}^+ :\geqslant Q_1$ and e_2 is $\widehat{\alpha}^+ :\geqslant Q_2$. Then $T \vdash P : e_1$ means $T \vdash P \geqslant Q_1$, and $T \vdash P : e_2$ means $T \vdash P \geqslant Q_2$. By the completeness of the least upper bound (lemma 73), $T \vDash Q_1 \vee Q_2 = Q$, and $T \vdash P \geqslant Q$. This way, ($\geqslant \&^+ \geqslant$) applies to infer $T \vdash e_1 \& e_2 = (\widehat{\alpha}^+ :\geqslant Q)$, and $T \vdash P : (\widehat{\alpha}^+ :\geqslant Q)$ holds by ($:\geqslant^{\text{SAT}}_+$).

Case 4. The negative cases are proved symmetrically.

□

Lemma 83 (Completeness of Constraint Merge). *Suppose that $\Sigma \vdash C_1 : \Upsilon_1$ and $\Sigma \vdash C_2 : \Upsilon_2$. If there exists a substitution $\Sigma \vdash \widehat{\sigma} : \Upsilon_1 \cup \Upsilon_2$ such that $\Sigma \vdash \widehat{\sigma} : C_1$ and $\Sigma \vdash \widehat{\sigma} : C_2$ then $\Sigma \vdash C_1 \& C_2 = C$ is defined.*

Proof. By definition, $C_1 \& C_2$ is a union of

(1) entries of C_1, which do not have matching entries in C_2,
(2) entries of C_2, which do not have matching entries in C_1, and
(3) the merge of matching entries.

This way, to show that $\Sigma \vdash C_1 \& C_2 = C$ is defined, we need to demonstrate that each of these components is defined and satisfies the required property (that the result of $\widehat{\sigma}$ satisfies the corresponding constraint entry).

It is clear that the first two components of this union exist. Moreover, if e is an entry of C_i restricting $\widehat{\alpha}^\pm \notin \textbf{dom}\,(C_2)$, then $\Sigma \vdash \widehat{\sigma} : C_i$ implies $\Sigma(\widehat{\alpha}^\pm) \vdash [\widehat{\sigma}]\widehat{\alpha}^\pm : e$,

Let us show that the third component exists. Let us take two entries $e_1 \in C_1$ and $e_2 \in C_2$ restricting the same variable $\widehat{\alpha}^\pm$. $\Sigma \vdash \widehat{\sigma} : C_1$ means that $\Sigma(\widehat{\alpha}^\pm) \vdash [\widehat{\sigma}]\widehat{\alpha}^\pm : e_1$ and $\Sigma \vdash \widehat{\sigma} : C_2$ means $\Sigma(\widehat{\alpha}^\pm) \vdash [\widehat{\sigma}]\widehat{\alpha}^\pm : e_2$. Then by lemma 82, $\Sigma(\widehat{\alpha}^\pm) \vdash e_1 \& e_2 = e$ is defined and $\Sigma(\widehat{\alpha}^\pm) \vdash [\widehat{\sigma}]\widehat{\alpha}^\pm : e$.

□

9.13 Negative Subtyping

Observation 13 (Negative Algorithmic Subtyping is Deterministic). *For fixed T, Σ, M, and N, if $T; \Sigma \vDash N \leqslant M \dashv C$ and $T; \Sigma \vDash N \leqslant M \dashv C'$ then $C = C'$.*

Proof. First, notice that the shape of the input uniquely determines the rule applied to infer $T; \Sigma \vDash N \leqslant M \dashv C$, which is consequently, the same rule used to infer $T; \Sigma \vDash N \leqslant M \dashv C'$.

Second, notice that for each of the inference rules, the premises are deterministic on the input. Specifically,

- ($\uparrow^{\leqslant}$) relies on unification, which is deterministic by observation 4;
- ($\forall^{\leqslant}$) relies on the choice of fresh algorithmic variables, which is deterministic as discussed in section 2.2, and on the negative subtyping, which is deterministic by the induction hypothesis;
- ($\rightarrow^{\leqslant}$) uses the negative subtyping (deterministic by the induction hypothesis), the positive subtyping (observation 10), and the merge of subtyping constraints (observation 12);

□

Lemma 84 (Soundness of Negative Subtyping). *If $T \vdash^{\supseteq} \Sigma$, $T \vdash M$, $T\,;\textbf{dom}\,(\Sigma) \vdash N$ and $T; \Sigma \vDash N \leqslant M \dashv C$, then $\Sigma \vdash C : \textbf{fav}(N)$ and for any normalized $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : C$, $T \vdash [\widehat{\sigma}]N \leqslant M$.*

Proof. We prove it by induction on $T; \Sigma \vDash N \leqslant M \dashv C$.

Suppose that $\widehat{\sigma}$ is normalized and $\Sigma \vdash \widehat{\sigma} : C$, Let us consider the last rule to infer this judgment.

Case 1. $(\to^{\leqslant})$. Then $T;\,\Sigma \vDash N \leqslant M \dashv C$ has shape $T;\,\Sigma \vDash P \to N' \leqslant Q \to M' \dashv C$
On the next step, the the algorithm makes two recursive calls: $T;\,\Sigma \vDash P \geqslant Q \dashv C_1$ and $T;\,\Sigma \vDash N' \leqslant M' \dashv C_2$ and returns $\Sigma \vdash C_1 \,\&\, C_2 = C$ as the result.
By the soundness of constraint merge (lemma 81), $\Sigma \vdash \widehat{\sigma} : C_1$ and $\Sigma \vdash \widehat{\sigma} : C_2$. Then by the soundness of positive subtyping (lemma 78), $T \vdash [\widehat{\sigma}]P \geqslant Q$; and by the induction hypothesis, $T \vdash [\widehat{\sigma}]N' \leqslant M'$. This way, by $(\to^{\leqslant})$, $T \vdash [\widehat{\sigma}](P \to N') \leqslant Q \to M'$.
Case 2. $(\mathrm{VAR}^{\leqslant}_{-})$, and then $T;\,\Sigma \vDash N \leqslant M \dashv C$ has shape $T;\,\Sigma \vDash \alpha^- \leqslant \alpha^- \dashv \cdot$
This case is symmetric to case 2 of lemma 78.
Case 3. $(\uparrow^{\leqslant})$, and then $T;\,\Sigma \vDash N \leqslant M \dashv C$ has shape $T;\,\Sigma \vDash \uparrow P \leqslant \uparrow Q \dashv C$
This case is symmetric to case 3 of lemma 78.
Case 4. $(\forall^{\leqslant})$, and then $T;\,\Sigma \vDash N \leqslant M \dashv C$ has shape $T;\,\Sigma \vDash \forall\overrightarrow{\alpha^+}.\, N' \leqslant \forall\overrightarrow{\beta^+}.\, M' \dashv C$ s.t. either $\overrightarrow{\alpha^+}$ or $\overrightarrow{\beta^+}$ is not empty
This case is symmetric to case 4 of lemma 78.

□

Lemma 85 (Completeness of the Negative Subtyping). *Suppose that* $T \vdash^{\supseteq} \Sigma$, $T \vdash M$, $T;\mathbf{dom}\,(\Sigma) \vdash N$, *and* N *does not contain negative unification variables* ($\widehat{\alpha}^- \notin \mathbf{fav}\,N$)*. Then for any* $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N)$ *such that* $T \vdash [\widehat{\sigma}]N \leqslant M$*, there exists* $T;\,\Sigma \vDash N \leqslant M \dashv C$ *and moreover,* $\Sigma \vdash \widehat{\sigma} : C$.

Proof. We prove it by induction on $T \vdash [\widehat{\sigma}]N \leqslant M$. Let us consider the last rule used in the derivation of $T \vdash [\widehat{\sigma}]N \leqslant M$.

Case 1. $T \vdash [\widehat{\sigma}]N \leqslant M$ is derived by $(\uparrow^{\leqslant})$
Then $N = \uparrow P$, since the substitution $[\widehat{\sigma}]N$ must preserve the top-level constructor of $N \neq \widehat{\alpha}^-$ (since by assumption, $\widehat{\alpha}^- \notin \mathbf{fav}\,N$), and $Q = \downarrow M$, and by inversion, $T \vdash [\widehat{\sigma}]N \simeq^{\leqslant} M$. The rest of the proof is symmetric to case 3 of lemma 79: notice that the algorithm does not make a recursive call, and the difference in the induction statement for the positive and the negative case here does not matter.
Case 2. $T \vdash [\widehat{\sigma}]N \leqslant M$ is derived by $(\to^{\leqslant})$, i.e. $[\widehat{\sigma}]N = [\widehat{\sigma}]P \to [\widehat{\sigma}]N'$ and $M = Q \to M'$, and by inversion, $T \vdash [\widehat{\sigma}]P \geqslant Q$ and $T \vdash [\widehat{\sigma}]N' \leqslant M'$.
The algorithm makes two recursive calls: $T;\,\Sigma \vDash P \geqslant Q \dashv C_1$ and $T;\,\Sigma \vDash N' \leqslant M' \dashv C_2$, and then returns $\Sigma \vdash C_1 \,\&\, C_2 = C$ as the result. Let us show that these recursive calls are successful and the returning constraints are fulfilled by $\widehat{\sigma}$.
Notice that from the inversion of $T \vdash M$, we have: $T \vdash Q$ and $T \vdash M'$; from the inversion of $T\,;\,\mathbf{dom}\,(\Sigma) \vdash N$, we have: $T\,;\,\mathbf{dom}\,(\Sigma) \vdash P$ and $T\,;\,\mathbf{dom}\,(\Sigma) \vdash N'$; and since N does not contain negative unification variables, N' does not contain negative unification variables either.
This way, we can apply the induction hypothesis to $T \vdash [\widehat{\sigma}]N' \leqslant M'$ to obtain $T;\,\Sigma \vDash N' \leqslant M' \dashv C_2$ such that $\Sigma \vdash C_2 : \mathbf{fav}(N')$ and $\Sigma \vdash \widehat{\sigma} : C_2$. Also, we can apply the completeness of the positive subtyping (lemma 79) to $T \vdash [\widehat{\sigma}]P \geqslant Q$ to obtain $T;\,\Sigma \vDash P \geqslant Q \dashv C_1$ such that $\Sigma \vdash C_1 : \mathbf{fav}(P)$ and $\Sigma \vdash \widehat{\sigma} : C_1$.
Finally, we need to show that the merge of C_1 and C_2 is successful and satisfies the required properties. To do so, we apply the completeness of subtyping constraint merge (lemma 83) (notice that $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P \to N')$ means $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P) \cup \mathbf{fav}(N')$). This way, $\Sigma \vdash C_1 \,\&\, C_2 = C$ is defined and $\Sigma \vdash \widehat{\sigma} : C$ holds.
Case 3. $T \vdash [\widehat{\sigma}]N \leqslant M$ is derived by $(\forall^{\leqslant})$. Since N does not contain negative unification variables, N must be of the form $\forall\overrightarrow{\alpha^+}.\, N'$, such that $[\widehat{\sigma}]N = \forall\overrightarrow{\alpha^+}.\, [\widehat{\sigma}]N'$ and $[\widehat{\sigma}]N' \neq \forall \ldots$ (assuming $\overrightarrow{\alpha^+}$ does not intersect with the range of $\widehat{\sigma}$). Also, $M = \forall\overrightarrow{\beta^+}.\, M'$ and either $\overrightarrow{\alpha^+}$ or $\overrightarrow{\beta^+}$ is non-empty.

The rest of the proof is symmetric to case 4 of lemma 79. To apply the induction hypothesis, we need to show additionally that there are no negative unification variables in $N_0 = [\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]N'$. This is because $\mathbf{fav}N_0 \subseteq \mathbf{fav}N \cup \overrightarrow{\widehat{\alpha}^+}$, and N is free of negative unification variables by assumption.

Case 4. $T \vdash [\widehat{\sigma}]N \leqslant M$ is derived by ($\text{Var}^{\leqslant}_{-}$).

Then $N = [\widehat{\sigma}]N = \alpha^- = M$. Here the first equality holds because N is not a unification variable: by assumption, N is free of negative unification variables. The second and the third equations hold because ($\text{Var}^{\leqslant}_{-}$) was applied.

The rest of the proof is symmetric to case 2 of lemma 79.

□

10 PROPERTIES OF THE DECLARATIVE TYPING

Definition 30 (Number of prenex quantifiers). *Let us define* $\text{npq}(N)$ *and* $\text{npq}(P)$ *as the number of prenex quantifiers in these types, i.e.*

- $+$ $\text{npq}(\exists\overrightarrow{\alpha^-}.\,P) = |\overrightarrow{\alpha^-}|$, *if* $P \neq \exists\overrightarrow{\beta^-}.\,P'$,
- $-$ $\text{npq}(\forall\overrightarrow{\alpha^+}.\,N) = |\overrightarrow{\alpha^+}|$, *if* $N \neq \forall\overrightarrow{\beta^+}.\,N'$.

Definition 31 (Size of a Declarative Judgement). *For a declarative typing judgment J let us define a metrics* $\text{size}(J)$ *as a pair of numbers in the following way:*

- $+$ $\text{size}(T;\Gamma \vdash v\colon P) = (\text{size}(v), 0)$;
- $-$ $\text{size}(T;\Gamma \vdash c\colon N) = (\text{size}(c), 0)$;
- $\bullet$ $\text{size}(T\,;\Gamma \vdash N \bullet \vec{v} \Rightarrow\!\!\!> M) = (\text{size}(\vec{v}), \text{npq}(N))$

where $\text{size}(v)$ *or* $\text{size}(c)$ *is the size of the syntax tree of the term v or c and* $\text{size}(\vec{v})$ *is the sum of sizes of the terms in* $\vec{v}$.

Definition 32 (Number of Equivalence Nodes). *For a tree T inferring a declarative typing judgment, let us define a function* $\text{eq\_nodes}(T)$ *as the number of nodes in T labeled with* ($\simeq^{\text{INF}}_{+}$) *or* ($\simeq^{\text{INF}}_{-}$).

Definition 33 (Metric). *For a tree T inferring a declarative typing judgment J, let us define a metric* $\text{metric}(T)$ *as a pair* $(\text{size}(J), \text{eq\_nodes}(T))$.

Lemma 49. *If* $T\,;\Gamma \vdash N_1 \bullet \vec{v} \Rightarrow\!\!\!> M$ *and* $T \vdash N_1 \simeq^{\leqslant} N_2$ *then* $T\,;\Gamma \vdash N_2 \bullet \vec{v} \Rightarrow\!\!\!> M$.

Proof. By lemma 32, $T \vdash N_1 \simeq^{\leqslant} N_2$ implies $N_1 \simeq^{D} N_2$. Let us prove the required judgement by induction on $N_1 \simeq^D N_2$. Let us consider the last rule used in the derivation.

Case 1. ($\text{Var}^{\simeq^D}_{-}$). It means that N_1 is α^- and N_2 is α^-. Then the required property coincides with the assumption.

Case 2. ($\uparrow^{\simeq^D}$). It means that N_1 is $\uparrow P_1$ and N_2 is $\uparrow P_2$. where $P_1 \simeq^D P_2$.

Then the only rule applicable to infer $T\,;\Gamma \vdash \uparrow P_1 \bullet \vec{v} \Rightarrow\!\!\!> M$ is ($\emptyset^{\text{INF}}_{\bullet\Rightarrow}$), meaning that $\vec{v} = \cdot$ and $T \vdash \uparrow P_1 \simeq^{\leqslant} M$. Then by transitivity of equivalence corollary 10, $T \vdash \uparrow P_2 \simeq^{\leqslant} M$, and then ($\emptyset^{\text{INF}}_{\bullet\Rightarrow}$) is applicable to infer $T\,;\Gamma \vdash \uparrow P_2 \bullet \cdot \Rightarrow\!\!\!> M$.

Case 3. ($\rightarrow^{\simeq^D}$). Then we are proving that $T\,;\Gamma \vdash (Q_1 \rightarrow N_1) \bullet v, \vec{v} \Rightarrow\!\!\!> M$ and $Q_1 \rightarrow N_1 \simeq^D Q_2 \rightarrow N_2$ imply $T\,;\Gamma \vdash (Q_2 \rightarrow N_2) \bullet v, \vec{v} \Rightarrow\!\!\!> M$.

By inversion, $(Q_1 \rightarrow N_1) \simeq^D (Q_2 \rightarrow N_2)$ means $Q_1 \simeq^D Q_2$ and $N_1 \simeq^D N_2$.

By inversion of $T\,;\Gamma \vdash (Q_1 \rightarrow N_1) \bullet v, \vec{v} \Rightarrow\!\!\!> M$:

1. $T;\Gamma \vdash v\colon P$
2. $T \vdash Q_1 \geqslant P$, and then by transitivity lemma 22, $T \vdash Q_2 \geqslant P$;
3. $T\,;\Gamma \vdash N_1 \bullet \vec{v} \Rightarrow\!\!\!> M$, and then by induction hypothesis, $T\,;\Gamma \vdash N_2 \bullet \vec{v} \Rightarrow\!\!\!> M$.

Since we have $T;\Gamma \vdash v\colon P$, $T \vdash Q_2 \geqslant P$ and $T\,;\Gamma \vdash N_2 \bullet \vec{v} \Rightarrow\!\!\!> M$, we can apply $(\to^{\text{INF}}_{\bullet\Rightarrow})$ to infer $T\,;\Gamma \vdash (Q_2 \to N_2) \bullet v, \vec{v} \Rightarrow\!\!\!> M$.

Case 4. $(\forall^{\simeq^D})$ Then we are proving that $T;\Gamma \vdash \forall\overrightarrow{\alpha^+}_1.\, N'_1 \bullet \vec{v} \Rightarrow\!\!\!> M$ and $\forall\overrightarrow{\alpha^+}_1.\, N'_1 \simeq^D \forall\overrightarrow{\alpha^+}_2.\, N'_2$ imply $T\,;\Gamma \vdash \forall\overrightarrow{\alpha^+}_2.\, N'_2 \bullet \vec{v} \Rightarrow\!\!\!> M$.

By inversion of $\forall\overrightarrow{\alpha^+}_1.\, N'_1 \simeq^D \forall\overrightarrow{\alpha^+}_2.\, N'_2$:

1. $\overrightarrow{\alpha^+}_2 \cap \mathbf{fv}\, N_1 = \emptyset$,
2. there exists a bijection $\mu : (\overrightarrow{\alpha^+}_2 \cap \mathbf{fv}\, N'_2) \leftrightarrow (\overrightarrow{\alpha^+}_1 \cap \mathbf{fv}\, N'_1)$ such that $N'_1 \simeq^D [\mu]N'_2$.

By inversion of $T\,;\Gamma \vdash \forall\overrightarrow{\alpha^+}_1.\, N'_1 \bullet \vec{v} \Rightarrow\!\!\!> M$:

1. $T \vdash \sigma : \overrightarrow{\alpha^+}_1$
2. $T\,;\Gamma \vdash [\sigma]N'_1 \bullet \vec{v} \Rightarrow\!\!\!> M$
3. $\vec{v} \neq \cdot$

Let us construct $T \vdash \sigma_0 : \overrightarrow{\alpha^+}_2$ in the following way:

$$\begin{cases} [\sigma_0]\alpha^+ = [\sigma][\mu]\alpha^+ & \text{if } \alpha^+ \in \overrightarrow{\alpha^+}_2 \cap \mathbf{fv}\, N'_2 \\ [\sigma_0]\alpha^+ = \exists\beta^-.\, {\downarrow}\beta^- & \text{otherwise (the type does not matter here)} \end{cases}$$

Then to infer $T\,;\Gamma \vdash N_2 \bullet \vec{v} \Rightarrow\!\!\!> M$, we apply $(\to^{\text{INF}}_{\bullet\Rightarrow})$ with σ_0. Let us show the required premises:

1. $T \vdash \sigma_0 : \overrightarrow{\alpha^+}_2$ by construction;
2. $\vec{v} \neq \cdot$ as noted above;
3. To show $T;\Gamma \vdash [\sigma_0]N'_2 \bullet \vec{v} \Rightarrow\!\!\!> M$, Notice that $[\sigma_0]N'_2 = [\sigma][\mu]N'_2$ and since $[\mu]N'_2 \simeq^D N'_1$, $[\sigma][\mu]N'_2 \simeq^D [\sigma]N'_1$. This way, by lemma 27, $T \vdash [\sigma]N'_1 \simeq^{\leqslant} [\sigma_0]N'_2$. Then the required judgement holds by the induction hypothesis applied to $T\,;\Gamma \vdash [\sigma]N'_1 \bullet \vec{v} \Rightarrow$ $> M$.

□

Lemma 48 (Declarative typing is preserved under context equivalence). *Assuming $T \vdash \Gamma_1$, $T \vdash \Gamma_2$, and $T \vdash \Gamma_1 \simeq^{\leqslant} \Gamma_2$:*

+ *for any tree T_1 inferring $T;\Gamma_1 \vdash v\colon P$, there exists a tree T_2 inferring $T;\Gamma_2 \vdash v\colon P$.*
− *for any tree T_1 inferring $T;\Gamma_1 \vdash c\colon N$, there exists a tree T_2 inferring $T;\Gamma_2 \vdash c\colon N$.*
• *for any tree T_1 inferring $T;\Gamma_1 \vdash N \bullet \vec{v} \Rightarrow\!\!\!> M$, there exists a tree T_2 inferring $T;\Gamma_2 \vdash N \bullet \vec{v} \Rightarrow\!\!\!> M$.*

Proof. Let us prove it by induction on the metric(T_1). Let us consider the last rule applied in T_1 (i.e., its root node).

Case 1. $(\text{Var}^{\text{INF}})$
Then we are proving that $T;\Gamma_1 \vdash x\colon P$ implies $T;\Gamma_2 \vdash x\colon P$. By inversion, $x : P \in \Gamma_1$, and since $T \vdash \Gamma_1 \simeq^{\leqslant} \Gamma_2$, $x : P' \in \Gamma_2$ for some P' such that $T \vdash P \simeq^{\leqslant} P'$. Then we infer $T;\Gamma_2 \vdash x\colon P'$ by $(\text{Var}^{\text{INF}})$, and next, $T;\Gamma_2 \vdash x\colon P$ by $(\simeq^{\text{INF}}_+)$.

Case 2. For $(\{\}^{\text{INF}})$, $(\text{ANN}^{\text{INF}}_+)$, (Λ^{INF}), $(\text{RET}^{\text{INF}})$, and $(\text{ANN}^{\text{INF}}_-)$ the proof is analogous. We apply the induction hypothesis to the premise of the rule to substitute Γ_1 for Γ_2 in it. The induction is applicable because the metric of the premises is less than the metric of the conclusion: the term in the premise is a syntactic subterm of the term in the conclusion.
And after that, we apply the same rule to infer the required judgment.

Case 3. $(\simeq^{\text{INF}}_+)$ and $(\simeq^{\text{INF}}_-)$ In these cases, the induction hypothesis is also applicable to the premise: although the first component of the metric is the same for the premise and the conclusion: $\text{size}(T;\Gamma \vdash c\colon N') = \text{size}(T;\Gamma \vdash c\colon N) = \text{size}(c)$, the second component of the metric is less for the premise by one, since the equivalence rule was applied to turn the premise tree into $T1$. Having made this note, we continue the proof in the same way as in the previous case.

Case 4. (λ^{INF}) Then we prove that $T; \Gamma_1 \vdash \lambda x : P.\ c \colon P \to N$ implies $T; \Gamma_2 \vdash \lambda x : P.\ c \colon P \to N$. Analogously to the previous cases, we apply the induction hypothesis to the equivalent contexts $T \vdash \Gamma_1, x : P \simeq^{\leqslant} \Gamma_2, x : P$ and the premise $T; \Gamma_1, x : P \vdash c \colon N$ to obtain $T; \Gamma_2, x : P \vdash c \colon N$. Notice that c is a subterm of $\lambda x : P.\ c$, i.e., the metric of the premise tree is less than the metric of the conclusion, and the induction hypothesis is applicable. Then we infer $T; \Gamma_2 \vdash \lambda x : P.\ c \colon P \to N$ by (λ^{INF}).

Case 5. $(\text{LET}^{\text{INF}})$ Then we prove that $T; \Gamma_1 \vdash \mathbf{let}\, x = v\, ; c \colon N$ implies $T; \Gamma_2 \vdash \mathbf{let}\, x = v\, ; c \colon N$. First, we apply the induction hypothesis to $T; \Gamma_1 \vdash v \colon P$ to obtain $T; \Gamma_2 \vdash v \colon P$ of the same pure size.
Then we apply the induction hypothesis to the equivalent contexts $T \vdash \Gamma_1, x : P \simeq^{\leqslant} \Gamma_2, x : P$ and the premise $T; \Gamma_1, x : P \vdash c \colon N$ to obtain $T; \Gamma_2, x : P \vdash c \colon N$. Then we infer $T; \Gamma_2 \vdash \mathbf{let}\, x = v\, ; c \colon N$ by $(\text{LET}^{\text{INF}})$.

Case 6. $(\text{LET}^{\text{INF}}_{@})$ Then we prove that $T; \Gamma_1 \vdash \mathbf{let}\, x = v(\vec{v})\, ; c \colon N$ implies $T; \Gamma_2 \vdash \mathbf{let}\, x = v(\vec{v})\, ; c \colon N$.
We apply the induction hypothesis to each of the premises. to rewrite:

- $T; \Gamma_1 \vdash v \colon \downarrow M$ into $T; \Gamma_2 \vdash v \colon \downarrow M$,
- $T\, ; \Gamma_1 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$ into $T\, ; \Gamma_2 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$.
- $T; \Gamma_1, x : Q \vdash c \colon N$ into $T; \Gamma_2, x : Q \vdash c \colon N$ (notice that $T \vdash \Gamma_1, x : Q \simeq^{\leqslant} \Gamma_2, x : Q$).

It is left to show the principality of $T\, ; \Gamma_2 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$. Let us assume that this judgment holds for other Q', i.e. there exists a tree T_0 inferring $T\, ; \Gamma_2 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q'$. Then notice that the induction hypothesis applies to T_0: the first component of the first component of metric(T_0) is $S = \sum_{v \in \vec{v}} \text{size}(v)$, and it is less than the corresponding component of metric(T_1), which is size$(\mathbf{let}\, x = v(\vec{v})\, ; c) = 1 + \text{size}(v) + \text{size}(c) + S$. This way, $T\, ; \Gamma_1 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q'$ holds by the induction hypothesis, but since $T\, ; \Gamma_1 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$ is principal, we have $T \vdash Q' \geqslant Q$. This way, $T\, ; \Gamma_2 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$ principal. Then we infer $T; \Gamma_2 \vdash \mathbf{let}\, x = v(\vec{v})\, ; c \colon N$ by $(\text{LET}^{\text{INF}}_{@})$.

Case 7. $(\text{LET}^{\text{INF}}_{:@})$ Then we prove that $T; \Gamma_1 \vdash \mathbf{let}\, x : P = v(\vec{v})\, ; c \colon N$ implies $T; \Gamma_2 \vdash \mathbf{let}\, x : P = v(\vec{v})\, ; c \colon N$.
As in the previous case, we apply the induction hypothesis to each of the premises and rewrite:

- $T; \Gamma_1 \vdash v \colon \downarrow M$ into $T; \Gamma_2 \vdash v \colon \downarrow M$,
- $T\, ; \Gamma_1 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$ into $T\, ; \Gamma_2 \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$, and
- $T; \Gamma_1, x : P \vdash c \colon N$ into $T; \Gamma_2, x : P \vdash c \colon N$ (notice that $T \vdash \Gamma_1, x : P \simeq^{\leqslant} \Gamma_2, x : P$).

Notice that $T \vdash P$ and $T \vdash \uparrow Q \leqslant \uparrow P$ do not depend on the variable context, and hold by assumption. Then we infer $T; \Gamma_2 \vdash \mathbf{let}\, x : P = v(\vec{v})\, ; c \colon N$ by $(\text{LET}^{\text{INF}}_{:@})$.

Case 8. $(\text{LET}^{\text{INF}}_{\exists})$, and $(\text{ANN}^{\text{INF}}_{-})$ are proved in the same way.

Case 9. $(\emptyset^{\text{INF}}_{\bullet\Rrightarrow})$ Then we are proving that $T\, ; \Gamma_1 \vdash N \bullet \cdot \Rrightarrow N'$ (inferred by $(\emptyset^{\text{INF}}_{\bullet\Rrightarrow})$) implies $T\, ; \Gamma_2 \vdash N \bullet \cdot \Rrightarrow N'$.
To infer $T\, ; \Gamma_2 \vdash N \bullet \cdot \Rrightarrow N'$, we apply $(\emptyset^{\text{INF}}_{\bullet\Rrightarrow})$, noting that $T \vdash N \simeq^{\leqslant} N'$ holds by assumption.

Case 10. $(\to^{\text{INF}}_{\bullet\Rrightarrow})$ Then we are proving that $T\, ; \Gamma_1 \vdash Q \to N \bullet v, \vec{v} \Rrightarrow M$ (inferred by $(\to^{\text{INF}}_{\bullet\Rrightarrow})$) implies $T\, ; \Gamma_2 \vdash Q \to N \bullet v, \vec{v} \Rrightarrow M$. And uniqueness of the M in the first case implies uniqueness in the second case.
By induction, we rewrite $T; \Gamma_1 \vdash v \colon P$ into $T; \Gamma_2 \vdash v \colon P$, and $T\, ; \Gamma_1 \vdash N \bullet \vec{v} \Rrightarrow M$ into $T\, ; \Gamma_2 \vdash N \bullet \vec{v} \Rrightarrow M$. Then we infer $T\, ; \Gamma_2 \vdash Q \to N \bullet v, \vec{v} \Rrightarrow M$ by $(\to^{\text{INF}}_{\bullet\Rrightarrow})$.

Now, let us show the uniqueness. The only rule that can infer $T\,;\Gamma_1 \vdash Q \to N \bullet v, \overrightarrow{v} \Rrightarrow M$ is $(\to^{\text{INF}}_{\bullet\Rrightarrow})$. Then by inversion, uniqueness of $T\,;\Gamma_1 \vdash Q \to N \bullet v, \overrightarrow{v} \Rrightarrow M$ implies uniqueness of $T\,;\Gamma_1 \vdash N \bullet \overrightarrow{v} \Rrightarrow M$. By the induction hypothesis, it implies the uniqueness of $T\,;\Gamma_2 \vdash N \bullet \overrightarrow{v} \Rrightarrow M$.

Suppose that $T\,;\Gamma_2 \vdash Q \to N \bullet v, \overrightarrow{v} \Rrightarrow M'$. By inversion, $T\,;\Gamma_2 \vdash N \bullet \overrightarrow{v} \Rrightarrow M'$, which by uniqueness of $T\,;\Gamma_2 \vdash N \bullet \overrightarrow{v} \Rrightarrow M$ implies $T \vdash M \simeq^{\leqslant} M'$.

Case 11. $(\forall^{\text{INF}}_{\bullet\Rrightarrow})$ Then we are proving that $T\,;\Gamma_1 \vdash \forall\overrightarrow{\alpha^+}.\, N \bullet \overrightarrow{v} \Rrightarrow M$ (inferred by $(\forall^{\text{INF}}_{\bullet\Rrightarrow})$) implies $T\,;\Gamma_2 \vdash \forall\overrightarrow{\alpha^+}.\, N \bullet \overrightarrow{v} \Rrightarrow M$.

By inversion, we have σ such that $T \vdash \sigma : \overrightarrow{\alpha^+}$ and $T\,;\Gamma_1 \vdash [\sigma]N \bullet \overrightarrow{v} \Rrightarrow M$ is inferred. Let us denote the inference tree as T_1'. Notice that the induction hypothesis is applicable to T_1': $\text{metric}(T_1') = ((\text{size}(\overrightarrow{v}), 0), x)$ is less than $\text{metric}(T_1) = ((\text{size}(\overrightarrow{v}), |\overrightarrow{\alpha^+}|), y)$ for any x and y, since $|\overrightarrow{\alpha^+}| > 0$ by inversion.

This way, by the induction hypothesis, there exists a tree T_2' inferring $T;\Gamma_2 \vdash [\sigma]N\bullet\overrightarrow{v} \Rrightarrow M$. Notice that the premises $\overrightarrow{v} \neq \cdot$, $T \vdash \sigma : \overrightarrow{\alpha^+}$, and $\overrightarrow{\alpha^+} \neq \cdot$ do not depend on the variable context, and hold by inversion. Then we infer $T\,;\Gamma_2 \vdash \forall\overrightarrow{\alpha^+}.\, N \bullet \overrightarrow{v} \Rrightarrow M$ by $(\forall^{\text{INF}}_{\bullet\Rrightarrow})$.

□

11 PROPERTIES OF THE ALGORITHMIC TYPING

11.1 Singularity and Minimal Instantiation

Lemma 86 (Soundness of Minimal Instantiation). *Suppose that $T \vdash^{\supseteq} \Sigma$, $\Sigma \vdash C$, and $T\,;\mathbf{dom}\,(\Sigma) \vdash P$. If P is C-minimized by $\widehat{\sigma}$ then*

- *$\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\,P$,*
- *$\Sigma \vdash \widehat{\sigma} : C$,*
- *$\widehat{\sigma}$ is normalized, and*
- *for any other $\Sigma \vdash \widehat{\sigma}' : \mathbf{fav}\,P$ respecting C (i.e., $\Sigma \vdash \widehat{\sigma}' : C$), we have $T \vdash [\widehat{\sigma}']P \geqslant [\widehat{\sigma}]P$.*

Proof. We prove it by induction on the inference of P is C-minimized by $\widehat{\sigma}$. Let us consider the last rule used in the inference.

Case 1. $(\text{UVar}^{\text{MIN}})$, which means that the inferred judgment is $\widehat{\alpha}^+$ is C-minimized by $(\mathbf{nf}\,(P)/\widehat{\alpha}^+)$, and by inversion, $(\widehat{\alpha}^+ :\geqslant P) \in C$. Let us show the required properties:
- $\Sigma \vdash (\mathbf{nf}\,(P)/\widehat{\alpha}^+) : \mathbf{fav}\widehat{\alpha}^+$ holds trivially;
- $\Sigma \vdash (\mathbf{nf}\,(P)/\widehat{\alpha}^+) : C$ holds since $T \vdash \mathbf{nf}\,(P) : (\widehat{\alpha}^+ :\geqslant P)$, which is true since $T \vdash \mathbf{nf}\,(P) \geqslant P$ by the soundness of normalization (lemma 39);
- $(\mathbf{nf}\,(P)/\widehat{\alpha}^+)$ is normalized trivially;
- let us take an arbitrary $\Sigma \vdash \widehat{\sigma}' : \widehat{\alpha}^+$ respecting C. Since $\widehat{\sigma}'$ respects C, $T \vdash [\widehat{\sigma}']\widehat{\alpha}^+ \geqslant P$ holds, and then $T \vdash [\widehat{\sigma}']\widehat{\alpha}^+ \geqslant \mathbf{nf}\,(P)$ holds by the soundness of normalization and transitivity of subtyping. Finally, $T \vdash [\widehat{\sigma}']\widehat{\alpha}^+ \geqslant \mathbf{nf}\,(P)$ can be rewritten as $T \vdash [\widehat{\sigma}']\widehat{\alpha}^+ \geqslant [(\mathbf{nf}\,(P)/\widehat{\alpha}^+)]\widehat{\alpha}^+$.

Case 2. $(\exists^{\text{MIN}})$, which means that the inferred judgment has form $\exists\overrightarrow{\alpha^-}.\, P$ is C-minimized by $\widehat{\sigma}$, and by inversion, P is C-minimized by $\widehat{\sigma}$. By applying the induction hypothesis to P is C-minimized by we have
- $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\,P$, which also means $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\exists\overrightarrow{\alpha^-}.\, P$,
- $\Sigma \vdash \widehat{\sigma} : C$,
- $\widehat{\sigma}$ is normalized, and
- for any other $\Sigma \vdash \widehat{\sigma}' : \mathbf{fav}\,P$ respecting C (i.e., $\Sigma \vdash \widehat{\sigma}' : C$), we have $T, \overrightarrow{\alpha^-} \vdash [\widehat{\sigma}']P \geqslant [\widehat{\sigma}]P$, which immediately implies $T \vdash [\widehat{\sigma}']\exists\overrightarrow{\alpha^-}.\, P \geqslant [\widehat{\sigma}]\exists\overrightarrow{\alpha^-}.\, P$ (the left-hand side

existential variables are instantiated with the corresponding right-hand side existential variables).

Case 3. ($\textsc{sing}^{\text{MIN}}$), which means that the inferred judgment has form P is C-minimized by $\widehat{\sigma}$, and by inversion, $\mathbf{fav}(P) \subseteq \mathbf{dom}\,(C)$ and $C|_{\mathbf{fav}(P)}$ **singular with** $\widehat{\sigma}$. Let us apply the soundness of singularity (lemma 90) to $C|_{\mathbf{fav}(P)}$ **singular with** $\widehat{\sigma}$ to obtain the following properties:

- $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P) \cap \mathbf{dom}\,(C)$, which also means $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(P)$,
- $\Sigma \vdash \widehat{\sigma} : C|_{\mathbf{fav}(P)}$,
- $\widehat{\sigma}$ is normalized, and
- for any other $\Sigma \vdash \widehat{\sigma}' : \mathbf{fav}(P)$ respecting $C|_{\mathbf{fav}(P)}$, we have $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}(P)$. The latter means that $T \vdash [\widehat{\sigma}']P \simeq^{\leqslant} [\widehat{\sigma}]P$, and in particular, $T \vdash [\widehat{\sigma}']P \geqslant [\widehat{\sigma}]P$.

$\square$

Lemma 87 (Completeness of Minimal Instantiation). *Suppose that $T \vdash^{\supseteq} \Sigma$, $\Sigma \vdash C$, T ; $\mathbf{dom}\,(\Sigma) \vdash P$, and there exists $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\,P$ respecting C ($\Sigma \vdash \widehat{\sigma} : C$) such that for any other $\Sigma \vdash \widehat{\sigma}' : \mathbf{fav}\,P$ respecting C ($\Sigma \vdash \widehat{\sigma}' : C$), we have $T \vdash [\widehat{\sigma}']P \geqslant [\widehat{\sigma}]P$. Then P is C-minimized by $\mathbf{nf}\,(\widehat{\sigma})$.*

Proof. We prove it by induction on P.

Case 1. $P = \widehat{\alpha}^+$. Suppose that $\widehat{\alpha}^+ \notin \mathbf{dom}\,(C)$. Then the instantiation of $\widehat{\alpha}^+$ is not restricted, and thus, any type can instantiate it. However, among unrestricted instantiations, there is no minimum: any type P is *not* a subtype of $\downarrow\uparrow P$, which contradicts the assumption. This way, $\widehat{\alpha}^+ \in \mathbf{dom}\,(C)$.

If the entry restricting $\widehat{\alpha}^+$ in C is a *subtyping* entry ($(\widehat{\alpha}^+ :\geqslant Q) \in C$), then we apply ($\text{UV}\textsc{ar}^{\text{MIN}}$) to infer $\widehat{\alpha}^+$ is C-minimized by $(\mathbf{nf}\,(Q)/\widehat{\alpha}^+)$. It is left to show that $\mathbf{nf}\,(Q) = \mathbf{nf}\,([\widehat{\sigma}]\widehat{\alpha}^+)$. Since $\Sigma \vdash \widehat{\sigma} : C$, and $(\widehat{\alpha}^+ :\geqslant Q) \in C$, we know that $T \vdash [\widehat{\sigma}]\widehat{\alpha}^+ \geqslant Q$. On the other hand, let us consider $\Sigma \vdash \widehat{\sigma}' : C$, that copies $\widehat{\sigma}$ on $\mathbf{dom}\,(C)$ except $\widehat{\alpha}^+$, where it is instantiated with Q. Then $T \vdash [\widehat{\sigma}']\widehat{\alpha}^+ \geqslant [\widehat{\sigma}]\widehat{\alpha}^+$ means $T \vdash Q \geqslant [\widehat{\sigma}]\widehat{\alpha}^+$, this way, $T \vdash Q \simeq^{\leqslant} [\widehat{\sigma}]\widehat{\alpha}^+$, which by lemma 46 means $\mathbf{nf}\,(Q) = \mathbf{nf}\,([\widehat{\sigma}]\widehat{\alpha}^+)$.

If the entry restricting $\widehat{\alpha}^+$ in C is an *equivalence* entry ($(\widehat{\alpha}^+ :\simeq Q) \in C$), then we wish to apply ($\textsc{sing}^{\text{MIN}}$). The first premise $\mathbf{fav}(\widehat{\alpha}^+) \subseteq \mathbf{dom}\,(C)$ holds by assumption; to infer $C|_{\widehat{\alpha}^+}$ **singular with** $\widehat{\sigma}_0$, we apply the completeness of singularity (lemma 91). It applies because all the substitutions satisfying $C|_{\widehat{\alpha}^+} = (\widehat{\alpha}^+ :\simeq Q)$ are equivalent on $\widehat{\alpha}^+$ by transitivity of equivalence (corollary 10): the satisfaction of this constraint means that the substitution sends $\widehat{\alpha}^+$ to Q or an equivalent type. This way, $C|_{\widehat{\alpha}^+}$ **singular with** $\widehat{\sigma}_0$ for some $\widehat{\sigma}_0$, which means $\widehat{\alpha}^+$ is C-minimized by $\widehat{\sigma}_0$. To show that $\widehat{\sigma}_0 = \mathbf{nf}\,(\widehat{\sigma})$ notice that Since $\widehat{\sigma}_0$ is normalized and equivalent to $\widehat{\sigma}$ on $\widehat{\alpha}^+$, and only has $\widehat{\alpha}^+$ in its domain (by soundness of singularity, lemma 90). This way, $\widehat{\alpha}^+$ is C-minimized by $\widehat{\sigma}$, as required.

Case 2. $P = \downarrow N$. Then since $T \vdash \downarrow[\widehat{\sigma}']N \geqslant \downarrow[\widehat{\sigma}]N$ means $T \vdash \downarrow[\widehat{\sigma}']N \simeq^{\leqslant} \downarrow[\widehat{\sigma}]N$ by inversion. Then by lemma 8, $\widehat{\sigma}$ is equivalent to any other substitution $\Sigma \vdash \widehat{\sigma}' : \mathbf{fav}\,N$ satisfying $C|_{\mathbf{fav}\,N}$, hence, the completeness of singularity (lemma 91) can be applied to conclude that

- $\mathbf{fav}(N) = \mathbf{dom}\,(C|_{\mathbf{fav}\,N})$, then $\mathbf{fav}(P) \subseteq \mathbf{dom}\,(C)$,
- $C|_{\mathbf{fav}\,N}$ **singular with** $\widehat{\sigma}_0$ for some (normalized) $\widehat{\sigma}_0$.

It means P is C-minimized by $\widehat{\sigma}_0$, and then since $\widehat{\sigma}_0$ is normalized and equivalent to $\widehat{\sigma}_0$ on $\mathbf{fav}(N)$, and its domain is $\mathbf{fav}(N)$, $\widehat{\sigma}_0 = \mathbf{nf}\,(\widehat{\sigma})$.

Case 3. $P = \exists\overrightarrow{\alpha^-}.\,\beta^+$ then as there are no algorithmic variables in P, $\mathbf{nf}\,([\widehat{\sigma}]P) = \beta^+$, and thus, we wish to show that $\exists\overrightarrow{\alpha^-}.\,\beta^+$ is C-minimized by $\cdot$. To do so, we apply ($\exists^{\text{MIN}}$), and it is left to show that β^+ is C-minimized by $\cdot$, which holds vacuously by ($\textsc{sing}^{\text{MIN}}$).

Case 4. $P = \exists\overrightarrow{\alpha^-}.\ \widehat{\alpha}^+$ then $T \vdash [\widehat{\sigma}']\exists\overrightarrow{\alpha^-}.\ \widehat{\alpha}^+ \geqslant [\widehat{\sigma}]\exists\overrightarrow{\alpha^-}.\ \widehat{\alpha}^+$ implies $T \vdash \exists\overrightarrow{\alpha^-}.\ [\widehat{\sigma}']\widehat{\alpha}^+ \geqslant \exists\overrightarrow{\alpha^-}.\ [\widehat{\sigma}]\widehat{\alpha}^+$ implies $T \vdash [\widehat{\sigma}']\widehat{\alpha}^+ \geqslant [\widehat{\sigma}]\widehat{\alpha}^+$. It means that $\widehat{\sigma}$ instantiates $\widehat{\alpha}^+$ to the minimal type among all the instantiations of $\widehat{\alpha}^+$ respecting C. In other words, we can apply the reasoning from case 1 to conclude that $\widehat{\alpha}^+$ is C-minimized by **nf** $(\widehat{\sigma})$. And then $(\exists^{\text{MIN}})$ gives us $\exists\overrightarrow{\alpha^-}.\ \widehat{\alpha}^+$ is C-minimized by **nf** $(\widehat{\sigma})$.

Case 5. $P = \exists\overrightarrow{\alpha^-}.\downarrow N$ then $T \vdash [\widehat{\sigma}']\exists\overrightarrow{\alpha^-}.\downarrow N \geqslant \exists\overrightarrow{\alpha^-}.\downarrow[\widehat{\sigma}]N$ implies $T \vdash \exists\overrightarrow{\alpha^-}.\downarrow[\widehat{\sigma}']N \geqslant \exists\overrightarrow{\alpha^-}.\downarrow[\widehat{\sigma}]N$ implies $T, \overrightarrow{\alpha^-} \vdash \downarrow[\sigma_0][\widehat{\sigma}']N \geqslant \downarrow[\widehat{\sigma}]N$ for some σ_0 implies $T, \overrightarrow{\alpha^-} \vdash [\sigma_0][\widehat{\sigma}']N \simeq^{\leqslant} [\widehat{\sigma}]N$. By lemma 8, it means in particular that $\widehat{\sigma}'$ and $\widehat{\sigma}$ are equivalent on **fav** N. This way, we can apply the completeness of singularity (lemma 91), and continue as in case 2 to conclude that $\downarrow N$ is C-minimized by **nf** $(\widehat{\sigma})$. Then by $(\exists^{\text{MIN}})$, we have $\exists\overrightarrow{\alpha^-}.\downarrow N$ is -minimized by **nf** $(\widehat{\sigma})$.

□

Observation 14 (Minimal Instantiation is Deterministic). *Suppose that* $T \vdash^{\supseteq} \Sigma, \Sigma \vdash C, T\,;\mathbf{dom}\,(\Sigma) \vdash P$*. Then* P *is* C*-minimized by* $\widehat{\sigma}$ *and* P *is* C*-minimized by* $\widehat{\sigma}'$ *implies* $\widehat{\sigma} = \widehat{\sigma}'$*.*

Proof. We prove it by induction on P is C-minimized by $\widehat{\sigma}$. It is easy to see that each inference rule is deterministic. □

Lemma 88 (Soundness of Entry Singularity).

+ *Suppose* e **singular with** P *for* P *well-formed in* T*. Then* $T \vdash P : e$*,* P *is normalized, and for any* $T \vdash P'$ *such that* $T \vdash P' : e$*,* $T \vdash P' \simeq^{\leqslant} P$*;*
– *Suppose* e **singular with** N *for* N *well-formed in* T*. Then* $T \vdash N : e$*,* N *is normalized, and for any* $T \vdash N'$ *such that* $T \vdash N' : e$*,* $T \vdash N' \simeq^{\leqslant} N$*.*

Proof. Let us consider how e **singular with** P or e **singular with** N is formed.

Case 1. $(\simeq^{\text{SING}}_{-})$, that is $e = \widehat{\alpha}^- :\simeq N_0$. and N is **nf** (N_0). Then $T \vdash N' : e$ means $T \vdash N' \simeq^{\leqslant} N_0$, (by inversion of $(:\simeq^{\text{SAT}}_{-})$), which by transitivity, using corollary 16, means $T \vdash N' \simeq^{\leqslant} \mathbf{nf}\,(N_0)$, as required.

Case 2. $(\simeq^{\text{SING}}_{+})$. This case is symmetric to the previous one.

Case 3. $(:\geqslant\alpha^{\text{SING}})$, that is $e = \widehat{\alpha}^+ :\geqslant \exists\overrightarrow{\alpha^-}.\ \beta^+$, and $P = \beta^+$.
Since $T \vdash \beta^+ \geqslant \exists\overrightarrow{\alpha^-}.\ \beta^+$, we have $T \vdash \beta^+ : e$, as required.
Notice that $T \vdash P' : e$ means $T \vdash P' \geqslant \exists\overrightarrow{\alpha^-}.\ \beta^+$. Let us show that it implies $T \vdash P' \simeq^{\leqslant} \beta^+$. By applying lemma 69 once, we have $T, \overrightarrow{\alpha^-} \vdash P' \geqslant \beta^+$. By applying it again, we notice that $T, \overrightarrow{\alpha^-} \vdash P' \geqslant \beta^+$ implies $P_i = \exists\overrightarrow{\alpha^-}'.\ \beta^+$. Finally, it is easy to see that $T \vdash \exists\overrightarrow{\alpha^-}'.\ \beta^+ \simeq^{\leqslant} \beta^+$

Case 4. $(:\geqslant\downarrow^{\text{SING}})$, that is $e = \widehat{\alpha}^+ :\geqslant \exists\overrightarrow{\beta^-}.\downarrow N_1$, where $N_1 \simeq^{D} {\beta^-}_j$, and $P = \exists\alpha^-.\downarrow\alpha^-$.
Since $T \vdash \exists\alpha^-.\downarrow\alpha^- \geqslant \exists\overrightarrow{\beta^-}.\downarrow N_1$ (by $(\exists^{\geqslant})$, with substitution N_1/α^-), we have $T \vdash \exists\alpha^-.\downarrow\alpha^- : e$, as required.
Notice $T \vdash P' : e$ means $T \vdash P' \geqslant \exists\overrightarrow{\beta^-}.\downarrow N_1$. Let us show that it implies $T \vdash P' \simeq^{\leqslant} \exists\alpha^-.\downarrow\alpha^-$.
$T \vdash P' \geqslant \exists\overrightarrow{\beta^-}.\downarrow N_1 \Rightarrow T \vdash \mathbf{nf}\,(P') \geqslant \exists\overrightarrow{\beta^-}'.\downarrow\mathbf{nf}\,(N_1)$

$$\begin{array}{rl}
(\text{where } \mathbf{ord}\,\overrightarrow{\beta^-}\ \mathbf{in}\ N' = \overrightarrow{\beta^-}') & \text{by corollary 17} \\
\Rightarrow T \vdash \mathbf{nf}\,(P') \geqslant \exists\overrightarrow{\beta^-}'.\downarrow\mathbf{nf}\,({\beta^-}_j) & \text{by lemma 42} \\
\Rightarrow T \vdash \mathbf{nf}\,(P') \geqslant \exists\overrightarrow{\beta^-}'.\downarrow{\beta^-}_n & \text{by definition of normalization} \\
\Rightarrow T \vdash \mathbf{nf}\,(P') \geqslant \exists{\beta^-}_j.\downarrow{\beta^-}_j & \text{since } \mathbf{ord}\,\overrightarrow{\beta^-}\ \mathbf{in}\ \mathbf{nf}\,(N_1) = {\beta^-}_j \\
\Rightarrow T, {\beta^-}_j \vdash \mathbf{nf}\,(P') \geqslant \downarrow{\beta^-}_j & \\
\text{and } {\beta^-}_j \notin \mathbf{fv}\,(\mathbf{nf}\,(P')) & \text{by lemma 70}
\end{array}$$

By lemma 70, the last subtyping means that $\mathbf{nf}\,(P') = \exists\overrightarrow{\alpha^-}.\downarrow N'$, such that

(1) $T, {\beta^-}_j, \overrightarrow{\alpha^-} \vdash N'$
(2) $\mathbf{ord}\ \overrightarrow{\alpha^-}\ \mathbf{in}\ N' = \overrightarrow{\alpha^-}$
(3) for some substitution $T, {\beta^-}_j \vdash \sigma : \overrightarrow{\alpha^-}$, $[\sigma]N' = {\beta^-}_j$.

Since ${\beta^-}_j \notin \mathbf{fv}\,(\mathbf{nf}\,(P'))$, the latter means that $N' = \alpha^-$, and then $\mathbf{nf}\,(P') = \exists\alpha^-. \downarrow\alpha^-$ for some α^-. Finally, notice that all the types of shape $\exists\alpha^-. \downarrow\alpha^-$ are equal.

□

Lemma 89 (Completeness of Entry Singularity)**.**

− *Suppose that there exists* N *well-formed in* T *such that for any* N' *well-formed in* T*,* $T \vdash N' : e$ *implies* $T \vdash N' \simeq^{\leqslant} N$*. Then* e **singular with nf** (N).

+ *Suppose that there exists* P *well-formed in* T *such that for any* P' *well-formed in* T*,* $T \vdash P' : e$ *implies* $T \vdash P' \simeq^{\leqslant} P$*. Then* e **singular with nf** (P) .

Proof.

− By lemma 76, there exists $T \vdash N' : e$. Since N' is negative, by inversion of $T \vdash N' : e$, e has shape $\widehat{\alpha}^- :\simeq M$, where $T \vdash N' \simeq^{\leqslant} M$, and transitively, $T \vdash N \simeq^{\leqslant} M$. Then $\mathbf{nf}\,(M) = \mathbf{nf}\,(N)$, and e **singular with nf** (M) (by $(\simeq_-^{\text{SING}})$) is rewritten as e **singular with nf** (N).

+ By lemma 76, there exists $T \vdash P' : e$, then by assumption, $T \vdash P' \simeq^{\leqslant} P$, which by lemma 77 implies $T \vdash P : e$.

Let us consider the shape of e:

Case 1. $e = (\widehat{\alpha}^+ :\simeq Q)$ then inversion of $T \vdash P : e$ implies $T \vdash P \simeq^{\leqslant} Q$, and hence, $\mathbf{nf}\,(P) = \mathbf{nf}\,(Q)$ (by lemma 46). Then e **singular with nf** (Q), which holds by $(\simeq_+^{\text{SING}})$, is rewritten as e **singular with nf** (P).

Case 2. $e = (\widehat{\alpha}^+ :\geqslant Q)$. Then the inversion of $T \vdash P : e$ implies $T \vdash P \geqslant Q$. Let us consider the shape of Q:

a. $Q = \exists\overrightarrow{\beta^-}. \beta^+$ (for potentially empty $\overrightarrow{\beta^-}$). Then $T \vdash P \geqslant \exists\overrightarrow{\beta^-}. \beta^+$ implies $T \vdash P \simeq^{\leqslant} \beta^+$ by lemma 69, as was noted in the proof of lemma 88, and hence, $\mathbf{nf}\,(P) = \beta^+$. Then e **singular with** β^+, which holds by $(:\geqslant\alpha^{\text{SING}})$, can be rewritten as e **singular with nf** (P).

b. $Q = \exists\overrightarrow{\beta^-}. \downarrow N$ (for potentially empty $\overrightarrow{\beta^-}$). Notice that $T \vdash \exists\gamma^-. \downarrow\gamma^- \geqslant \exists\overrightarrow{\beta^-}. \downarrow N$ (by $(\exists^{\geqslant})$), with substitution N/γ^-), and thus, $T \vdash \exists\gamma^-. \downarrow\gamma^- : e$ by $(:\geqslant_+^{\text{SAT}})$. Then by assumption, $T \vdash \exists\gamma^-. \downarrow\gamma^- \simeq^{\leqslant} P$, that is $\mathbf{nf}\,(P) = \exists\gamma^-. \downarrow\gamma^-$. To apply $(:\geqslant\downarrow^{\text{SING}})$ to infer $(\widehat{\alpha}^+ :\geqslant \exists\overrightarrow{\beta^-}. \downarrow N)$ **singular with** $\exists\gamma^-. \downarrow\gamma^-$, it is left to show that $N \simeq^D {\beta^-}_i$ for some i.

Since $T \vdash Q : e$, by assumption, $T \vdash Q \simeq^{\leqslant} P$, and by transitivity, $T \vdash Q \simeq^{\leqslant} \exists\gamma^-. \downarrow\gamma^-$. It implies $\mathbf{nf}\,(\exists\overrightarrow{\beta^-}. \downarrow N) = \exists\gamma^-. \downarrow\gamma^-$ (by lemma 46), which by definition of normalization means $\exists\overrightarrow{\beta^-}'. \downarrow\mathbf{nf}\,(N) = \exists\gamma^-. \downarrow\gamma^-$, where $\mathbf{ord}\ \overrightarrow{\beta^-}\ \mathbf{in}\ N' = \overrightarrow{\beta^-}'$. This way, $\overrightarrow{\beta^-}'$ is a variable β^-, and $\mathbf{nf}\,(N) = \beta^-$. Notice that $\beta^- \in \overrightarrow{\beta^-}' \subseteq \overrightarrow{\beta^-}$ by lemma 33. This way, $N \simeq^D \beta^-$ for $\beta^- \in \overrightarrow{\beta^-}$ (by lemma 46),

□

Lemma 90 (Soundness of Singularity)**.** *Suppose* $\Sigma \vdash C : \Upsilon$*, and* C **singular with** $\widehat{\sigma}$*. Then* $\Sigma \vdash \widehat{\sigma} : \Upsilon$*,* $\Sigma \vdash \widehat{\sigma} : C$*,* $\widehat{\sigma}$ *is normalized, and for any* $\widehat{\sigma}'$ *such that* $\Sigma \vdash \widehat{\sigma} : C$*,* $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma} : \Upsilon$*.*

Proof. Suppose that $\Sigma \vdash \widehat{\sigma}' : C$. It means that for every $e \in C$ restricting $\widehat{\alpha}^{\pm}$, $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}']\widehat{\alpha}^{\pm} : e$ holds. C **singular with** $\widehat{\sigma}$ means e **singular with** $[\widehat{\sigma}]\widehat{\alpha}^{\pm}$, and hence, by lemma 89, $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}']\widehat{\alpha}^{\pm} \simeq^{\leqslant} [\widehat{\sigma}]\widehat{\alpha}^{\pm}$ holds.

Since the uniqueness holds for every variable from $\mathbf{dom}\,(C)$, $\widehat{\sigma}$ is equivalent to $\widehat{\sigma}'$ on this set. □

Observation 15 (Singularity is Deterministic). *For a fixed C such that $\Sigma \vdash C : \Upsilon$, if C* **singular with** $\widehat{\sigma}$ *and C* **singular with** $\widehat{\sigma}'$*, then $\widehat{\sigma} = \widehat{\sigma}'$.*

Proof. By lemma 90, $\Sigma \vdash \widehat{\sigma} : \Upsilon$ and $\Sigma \vdash \widehat{\sigma}' : \Upsilon$. It means that both $\widehat{\sigma}$ and $\widehat{\sigma}'$ act as identity outside of Υ.

Moreover, for any $\widehat{\alpha}^{\pm} \in \Upsilon$, $\Sigma \vdash C : \Upsilon$ means that there is a unique $e \in C$ restricting $\widehat{\alpha}^{\pm}$. Then C **singular with** $\widehat{\sigma}$ means that e **singular with** $[\widehat{\sigma}]\widehat{\alpha}^{\pm}$. By looking at the inference rules, it is easy to see that $[\widehat{\sigma}]\widehat{\alpha}^{\pm}$ is uniquely determined by e, which, Similarly, $[\widehat{\sigma}']\widehat{\alpha}^{\pm}$ is also uniquely determined by e, in the same way, and hence, $[\widehat{\sigma}]\widehat{\alpha}^{\pm} = [\widehat{\sigma}']\widehat{\alpha}^{\pm}$. □

Lemma 91 (Completeness of Singularity). *For a given $\Sigma \vdash C$, suppose that all the substitutions satisfying C are equivalent on $\Upsilon \supseteq$* **dom** (C)*. In other words, suppose that there exists $\Sigma \vdash \widehat{\sigma}_1 : \Upsilon$ such that for any $\Sigma \vdash \widehat{\sigma} : \Upsilon$, $\Sigma \vdash \widehat{\sigma} : C$ implies $\Sigma \vdash \widehat{\sigma} \simeq^{\leqslant} \widehat{\sigma}_1 : \Upsilon$. Then*

- *C* **singular with** $\widehat{\sigma}_0$ *for some $\widehat{\sigma}_0$ and*
- $\Upsilon =$ **dom** (C).

Proof. First, let us assume $\Upsilon \neq$ **dom** (C). Then there exists $\widehat{\alpha}^{\pm} \in \Upsilon \setminus$ **dom** (C). Let us take $\Sigma \vdash \widehat{\sigma}_1 : \Upsilon$ such that any other substitution $\Sigma \vdash \widehat{\sigma} : \Upsilon$ satisfying C is equivalent to $\widehat{\sigma}_1$ on Υ.

Notice that $\Sigma \vdash \widehat{\sigma}_1 : C$: by lemma 76, there exists $\widehat{\sigma}'$ such that $\Sigma \vdash \widehat{\sigma}' : \Upsilon$ and $\Sigma \vdash \widehat{\sigma}' : C$, and by assumption, $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma}_1 : \Upsilon$, implying $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma}_1 :$ **dom** (C).

Let us construct $\widehat{\sigma}_2$ such that $\Sigma \vdash \widehat{\sigma}_2 : \Upsilon$ as follows:

$$\begin{cases} [\widehat{\sigma}_2]\widehat{\beta}^{\pm} = [\widehat{\sigma}_1]\widehat{\beta}^{\pm} & \text{if } \widehat{\beta}^{\pm} \neq \widehat{\alpha}^{\pm} \\ [\widehat{\sigma}_2]\widehat{\alpha}^{\pm} = T & \text{where } T \text{ is any closed type not equivalent to } [\widehat{\sigma}_1]\widehat{\alpha}^{\pm} \end{cases}$$

It is easy to see that $\Sigma \vdash \widehat{\sigma}_2 : C$ since $\widehat{\sigma}_1|_{\mathbf{dom}\,(C)} = \widehat{\sigma}_2|_{\mathbf{dom}\,(C)}$, and $\Sigma \vdash \widehat{\sigma}_1 : C$. However, $\Sigma \vdash \widehat{\sigma}_2 \simeq^{\leqslant} \widehat{\sigma}_1 : \Upsilon$ does not hold because by construction, $\Sigma(\widehat{\alpha}^{\pm}) \vdash [\widehat{\sigma}_2]\widehat{\alpha}^{\pm} \simeq^{\leqslant} [\widehat{\sigma}_1]\widehat{\alpha}^{\pm}$ does not hold. This way, we have a contradiction.

Second, let us show C **singular with** $\widehat{\sigma}_0$. Let us take arbitrary $e \in C$ restricting $\widehat{\beta}^{\pm}$. We need to show that e is singular. Notice that $\Sigma \vdash \widehat{\sigma}_1 : C$ implies $\Sigma(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}_1]\widehat{\beta}^{\pm}$ and $\Sigma(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}_1]\widehat{\beta}^{\pm} : e$. We will show that any other type satisfying e is equivalent to $[\widehat{\sigma}_1]\widehat{\beta}^{\pm}$, then by lemma 89, e **singular with** $[\widehat{\sigma}_1]\widehat{\beta}^{\pm}$.

- if $\widehat{\beta}^{\pm}$ is positive, let us take any type $\Sigma(\widehat{\beta}^{\pm}) \vdash P'$ and assume $\Sigma(\widehat{\beta}^{\pm}) \vdash P' : e$. We will show that $\Sigma(\widehat{\beta}^{\pm}) \vdash P' \simeq^{\leqslant} [\widehat{\sigma}_1]\widehat{\beta}^{\pm}$, which by lemma 46 will imply e **singular with nf** $([\widehat{\sigma}_1]\widehat{\beta}^{\pm})$.
  Let us construct $\widehat{\sigma}_2$ such that $\Sigma \vdash \widehat{\sigma}_2 : \Upsilon$ as follows:
  $$\begin{cases} [\widehat{\sigma}_2]\widehat{\gamma}^{\pm} = [\widehat{\sigma}_1]\widehat{\gamma}^{\pm} & \text{if } \widehat{\gamma}^{\pm} \neq \widehat{\beta}^{\pm} \\ [\widehat{\sigma}_2]\widehat{\beta}^{\pm} = P' \end{cases}$$
  It is easy to see that $\Sigma \vdash \widehat{\sigma}_2 : C$: for e, $\Sigma(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}_2]\widehat{\beta}^{\pm} : e$ by construction, since $\Sigma(\widehat{\beta}^{\pm}) \vdash P' : e$; for any other $e' \in C$ restricting $\widehat{\gamma}^{\pm}$, $[\widehat{\sigma}_2]\widehat{\gamma}^{\pm} = [\widehat{\sigma}_1]\widehat{\gamma}^{\pm}$, and $\Sigma(\widehat{\gamma}^{\pm}) \vdash [\widehat{\sigma}_1]\widehat{\gamma}^{\pm} : e'$ since $\Sigma \vdash \widehat{\sigma}_1 : C$. Then by assumption, $\Sigma \vdash \widehat{\sigma}_2 \simeq^{\leqslant} \widehat{\sigma}_1 : \Upsilon$, which in particular means $\Sigma(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}_2]\widehat{\beta}^{\pm} \simeq^{\leqslant} [\widehat{\sigma}_1]\widehat{\beta}^{\pm}$, that is $\Sigma(\widehat{\beta}^{\pm}) \vdash P' \simeq^{\leqslant} [\widehat{\sigma}_1]\widehat{\beta}^{\pm}$.
- if $\widehat{\beta}^{\pm}$ is negative, the proof is analogous.

□

11.2 Correctness of the Typing Algorithm

Lemma 92 (Determinacy of typing algorithm). *Suppose that $T \vdash \Gamma$ and $T \vdash^{\supseteq} \Sigma$. Then*

+ *If $T; \Gamma \models v \colon P$ and $T; \Gamma \models v \colon P'$ then $P = P'$.*

– *If* $T;\Gamma \vDash c\colon N$ *and* $T;\Gamma \vDash c\colon N'$ *then* $N = N'$.

• *If* $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M \dashv \Sigma'\,;C$ *and* $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M' \dashv \Sigma'\,;C'$ *then* $M = M'$, $\Sigma = \Sigma'$, *and* $C = C'$.

Proof. We show it by structural induction on the inference tree. Notice that the last rule used to infer the judgement is uniquely determined by the input, and that each premise of each inference rule is deterministic by the corresponding observation. □

Let us extend the declarative typing metric (definition 33) to the algorithmic typing.

Definition 34 (Size of an Algorithmic Judgement). *For an algorithmic typing judgement J let us define a metrics* $\text{size}(J)$ *as a pair of numbers in the following way:*

+ $\text{size}(T;\Gamma \vDash v\colon P) = (\text{size}(v), 0)$;
– $\text{size}(T;\Gamma \vDash c\colon N) = (\text{size}(c), 0)$;
• $\text{size}(T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M \dashv \Sigma'\,;C) = (\text{size}(\vec{v}), \text{npq}(N)))$

Definition 35 (Metric). *We extend the metric from definition 33 to the algorithmic typing in the following way. For a tree T inferring an algorithmic typing judgement J, we define* $\text{metric}(T)$ *as* $(\text{size}(J), 0)$.

Soundness and completeness are proved by mutual induction on the metric of the inference tree.

Lemma 93 (Soundness of typing). *Suppose that $T \vdash \Gamma$. For an inference tree T_1,*

+ *If T_1 infers $T;\Gamma \vDash v\colon P$ then $T \vdash P$ and $T;\Gamma \vdash v\colon P$*
– *If T_1 infers $T;\Gamma \vDash c\colon N$ then $T \vdash N$ and $T;\Gamma \vdash c\colon N$*
• *If T_1 infers $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M \dashv \Sigma'\,;C$ for $T \vdash^{\supseteq} \Sigma$ and $T\,;\mathbf{dom}\,(\Sigma) \vdash N$ free from negative algorithmic variables, then*
  (1) $T \vdash^{\supseteq} \Sigma'$
  (2) $\Sigma \subseteq \Sigma'$
  (3) $T\,;\mathbf{dom}\,(\Sigma') \vdash M$
  (4) $\mathbf{dom}\,(\Sigma) \cap \mathbf{fav}(M) \subseteq \mathbf{fav}\,N$
  (5) M *is normalized and free from negative algorithmic variables*
  (6) $\Sigma'|_{\mathbf{fav}\,N \cup \mathbf{fav}\,M} \vdash C$
  (7) for any $\Sigma' \vdash \widehat{\sigma} : \mathbf{fav}\,N \cup \mathbf{fav}\,M$, $\Sigma' \vdash \widehat{\sigma} : C$ *implies* $T\,;\Gamma \vdash [\widehat{\sigma}]\,N \bullet \vec{v} \Rrightarrow [\widehat{\sigma}]\,M$

Proof. We prove it by induction on $\text{metric}(T_1)$, mutually with the completeness of typing (lemma 93). Let us consider the last rule used to infer the derivation.

Case 1. (Var^{INF}) We are proving that if $T;\Gamma \vDash x\colon \mathbf{nf}\,(P)$ then $T \vdash \mathbf{nf}\,(P)$ and $T;\Gamma \vdash x\colon \mathbf{nf}\,(P)$.
By inversion, $x : P \in \Gamma$. Since $T \vdash \Gamma$, we have $T \vdash P$, and by corollary 14, $T \vdash \mathbf{nf}\,(P)$.
By applying (Var^{INF}) to $x : P \in \Gamma$, we infer $T;\Gamma \vdash x\colon P$. Finally, by ($\simeq^{\text{INF}}_{+}$), since $T \vdash P \simeq^{\leqslant} \mathbf{nf}\,(P)$ (corollary 16), we have $T;\Gamma \vdash x\colon \mathbf{nf}\,(P)$.

Case 2. ($\{\}^{\text{INF}}$)
We are proving that if $T;\Gamma \vDash \{c\}\colon {\downarrow}N$ then $T \vdash {\downarrow}N$ and $T;\Gamma \vdash \{c\}\colon {\downarrow}N$.
By inversion of $T;\Gamma \vDash \{c\}\colon {\downarrow}N$, we have $T;\Gamma \vDash c\colon N$. By the induction hypothesis applied to $T;\Gamma \vDash c\colon N$, we have
(1) $T \vdash N$, and hence, $T \vdash {\downarrow}N$;
(2) $T;\Gamma \vdash c\colon N$, which by ($\{\}^{\text{INF}}$) implies $T;\Gamma \vdash \{c\}\colon {\downarrow}N$.

Case 3. (ret^{INF}) The proof is symmetric to the previous case (case 2).

Case 4. ($\text{ann}^{\text{INF}}_{+}$) We are proving that if $T;\Gamma \vDash (v : Q)\colon \mathbf{nf}\,(Q)$ then $T \vdash \mathbf{nf}\,(Q)$ and $T;\Gamma \vdash (v : Q)\colon \mathbf{nf}\,(Q)$.
By inversion of $T;\Gamma \vDash (v : Q)\colon \mathbf{nf}\,(Q)$, we have:

(1) $T \vdash (v : Q)$, hence, $T \vdash Q$, and by corollary 14, $T \vdash \mathbf{nf}\,(Q)$;
(2) $T; \Gamma \vDash v\colon P$, which by the induction hypothesis implies $T \vdash P$ and $T; \Gamma \vdash v\colon P$;
(3) $T;\cdot \vDash Q \geqslant P \dashv \cdot$, which by lemma 78 implies $T \vdash [\cdot]Q \geqslant P$, that is $T \vdash Q \geqslant P$.

To infer $T; \Gamma \vdash (v : Q)\colon Q$, we apply $(\text{ANN}^{\text{INF}}_{+})$ to $T; \Gamma \vdash v\colon P$ and $T \vdash Q \geqslant P$. Then by $(\simeq^{\text{INF}}_{+})$, $T; \Gamma \vdash (v : Q)\colon \mathbf{nf}\,(Q)$.

Case 5. $(\text{ANN}^{\text{INF}}_{-})$ The proof is symmetric to the previous case (case 4).

Case 6. (λ^{INF}) We are proving that if $T; \Gamma \vDash \lambda x : P.\ c\colon \mathbf{nf}\,(P \to N)$ then $T \vdash \mathbf{nf}\,(P \to N)$ and $T; \Gamma \vdash \lambda x : P.\ c\colon \mathbf{nf}\,(P \to N)$.
By inversion of $T; \Gamma \vDash \lambda x : P.\ c\colon \mathbf{nf}\,(P \to N)$, we have $T \vdash \lambda x : P.\ c$, which implies $T \vdash P$.
Also by inversion of $T; \Gamma \vDash \lambda x : P.\ c\colon \mathbf{nf}\,(P \to N)$, we have $T; \Gamma, x : P \vDash c\colon N$, applying induction hypothesis to which gives us:
(1) $T \vdash N$, thus $T \vdash P \to N$, and by corollary 14, $T \vdash \mathbf{nf}\,(P \to N)$;
(2) $T; \Gamma, x : P \vdash c\colon N$, which by (λ^{INF}) implies $T; \Gamma \vdash \lambda x : P.\ c\colon P \to N$, and by $(\simeq^{\text{INF}}_{+})$, $T; \Gamma \vdash \lambda x : P.\ c\colon \mathbf{nf}\,(P \to N)$.

Case 7. (Λ^{INF}) We are proving that if $T; \Gamma \vDash \Lambda\alpha^+.\ c\colon \mathbf{nf}\,(\forall\alpha^+.\ N)$ then $T; \Gamma \vdash \Lambda\alpha^+.\ c\colon \mathbf{nf}\,(\forall\alpha^+.\ N)$ and $T \vdash \mathbf{nf}\,(\forall\alpha^+.\ N)$.
By inversion of $T, \alpha^+; \Gamma \vDash c\colon N$, we have $T \vdash \Lambda\alpha^+.\ c$, which implies $T, \alpha^+ \vdash c$.
Also by inversion of $T, \alpha^+; \Gamma \vDash c\colon N$, we have $T, \alpha^+; \Gamma \vDash c\colon N$. Obtaining the induction hypothesis to $T, \alpha^+; \Gamma \vDash c\colon N$, we have:
(1) $T, \alpha^+ \vdash N$, thus $T \vdash \forall\alpha^+.\ N$, and by corollary 14, $T \vdash \mathbf{nf}\,(\forall\alpha^+.\ N)$;
(2) $T, \alpha^+; \Gamma \vdash c\colon N$, which by (Λ^{INF}) implies $T; \Gamma \vdash \Lambda\alpha^+.\ c\colon \forall\alpha^+.\ N$, and by $(\simeq^{\text{INF}}_{+})$, $T; \Gamma \vdash \Lambda\alpha^+.\ c\colon \mathbf{nf}\,(\forall\alpha^+.\ N)$.

Case 8. $(\text{LET}^{\text{INF}})$ We are proving that if $T; \Gamma \vDash \mathbf{let}\, x = v\,; c\colon N$ then $T; \Gamma \vdash \mathbf{let}\, x = v\,; c\colon N$ and $T \vdash N$.
By inversion of $T; \Gamma \vDash \mathbf{let}\, x = v\,; c\colon N$, we have:
(1) $T \vdash \mathbf{let}\, x = v\,; c$, which gives us $T \vdash v$ and $T \vdash c$.
(2) $T; \Gamma \vDash v\colon P$, which by the induction hypothesis implies $T \vdash P$ (and thus, $T \vdash \Gamma, x : P$) and $T; \Gamma \vdash v\colon P$;
(3) $T; \Gamma, x : P \vDash c\colon N$, which by the induction hypothesis implies $T \vdash N$ and $T; \Gamma, x : P \vdash c\colon N$.

This way, $T; \Gamma \vdash \mathbf{let}\, x = v\,; c\colon N$ holds by $(\text{LET}^{\text{INF}})$.

Case 9. $(\text{LET}^{\text{INF}}_{:@})$ We are proving that if $T; \Gamma \vDash \mathbf{let}\, x : P = v(\vec{v})\,; c'\colon N$ then $T; \Gamma \vdash \mathbf{let}\, x : P = v(\vec{v})\,; c'\colon N$ and $T \vdash N$.
By inversion, we have:
(1) $T \vdash P$, hence, $T \vdash \Gamma, x : P$
(2) $T; \Gamma \vDash v\colon \downarrow M$
(3) $T\,; \Gamma\,; \cdot \vDash M \bullet \vec{v} \Longrightarrow \uparrow Q \dashv \Sigma\,; C_1$
(4) $T;\ \Sigma \vDash \uparrow Q \leqslant \uparrow P \dashv C_2$
(5) $\Sigma \vdash C_1 \,\&\, C_2 = C$
(6) $T; \Gamma, x : P \vDash c'\colon N$

By the induction hypothesis applied to $T; \Gamma \vDash v\colon \downarrow M$, we have $T; \Gamma \vdash v\colon \downarrow M$ and $T \vdash \downarrow M$ (and hence, $T\,; \mathbf{dom}\,(\Sigma) \vdash M$).
By the induction hypothesis applied to $T; \Gamma, x : P \vDash c'\colon N$, we have $T; \Gamma, x : P \vdash c'\colon N$ and $T \vdash N$.
By the induction hypothesis applied to $T\,; \Gamma\,; \cdot \vDash M \bullet \vec{v} \Longrightarrow \uparrow Q \dashv \Sigma\,; C_1$, we have:
(1) $T \vdash^{\supseteq} \Sigma$,
(2) $T\,; \mathbf{dom}\,(\Sigma) \vdash \uparrow Q$,

(3) $\Sigma'|_{\mathbf{fav}\, M \cup \mathbf{fav}\, Q} \vdash C_1$, and thus, $\mathbf{dom}\,(C_1) \subseteq \mathbf{fav}\, M \cup \mathbf{fav}\, Q$.
(4) for any $\Sigma' \vdash \widehat{\sigma} : C_1$, we have $T\,;\Gamma \vdash [\widehat{\sigma}] M \bullet \vec{v} \Rightarrow\!\!\!> [\widehat{\sigma}]{\uparrow}Q$.

By soundness of negative subtyping (lemma 84) applied to $T;\ \Sigma \vDash {\uparrow}Q \leqslant {\uparrow}P \dashv C_2$, we have $\Sigma \vdash C_2 : \mathbf{fav}({\uparrow}Q)$, and thus, $\mathbf{fav}({\uparrow}Q) = \mathbf{dom}\,(C_2)$.

By soundness of constraint merge (lemma 81), $\mathbf{dom}\,(C) = \mathbf{dom}\,(C_1) \cup \mathbf{dom}\,(C_2) \subseteq \mathbf{fav}\, M \cup \mathbf{fav}\, Q$ Then by lemma 76, let us take $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(M) \cup \mathbf{fav}(Q)$ and $\Sigma \vdash \widehat{\sigma} : C$. By the soundness of constraint merge, $\Sigma \vdash \widehat{\sigma} : C_1$ and $\Sigma \vdash \widehat{\sigma} : C_2$, and by weakening, $\Sigma' \vdash \widehat{\sigma} : C_1$ and $\Sigma' \vdash \widehat{\sigma} : C_2$.

Then as noted above (4), $T\,;\Gamma \vdash M \bullet \vec{v} \Rightarrow\!\!\!> [\widehat{\sigma}]{\uparrow}Q$ And again, by soundness of negative subtyping (lemma 84) applied to $T;\ \Sigma \vDash {\uparrow}Q \leqslant {\uparrow}P \dashv C_2$, we have $T \vdash [\widehat{\sigma}]{\uparrow}Q \leqslant {\uparrow}P$.

To infer $T;\Gamma \vdash \mathbf{let}\, x : P = v(\vec{v})\,;c' : N$, we apply the corresponding declarative rule ($\text{LET}^{\text{INF}}_{:@}$), where Q is $[\widehat{\sigma}]\, Q$. Notice that all the premises were already shown to hold above:
(1) $T \vdash P$ and $T;\Gamma \vdash v\colon {\downarrow}M$ from the assumption,
(2) $T\,;\Gamma \vdash M \bullet \vec{v} \Rightarrow\!\!\!> {\uparrow}[\widehat{\sigma}]\,Q$ holds since $[\widehat{\sigma}]{\uparrow}Q = {\uparrow}[\widehat{\sigma}]\,Q$,
(3) $T \vdash {\uparrow}[\widehat{\sigma}]\,Q \leqslant {\uparrow}P$ by soundness of negative subtyping,
(4) $T;\Gamma, x : P \vdash c' : N$ from the the induction hypothesis.

Case 10. ($\text{LET}^{\text{INF}}_{@}$) We are proving that if $T;\Gamma \vDash \mathbf{let}\, x = v(\vec{v})\,;c' : N$ then $T;\Gamma \vdash \mathbf{let}\, x = v(\vec{v})\,;c' : N$ and $T \vdash N$.

By the inversion, we have:
(1) $T;\Gamma \vDash v\colon {\downarrow}M$,
(2) $T\,;\Gamma\,;\cdot \vDash M \bullet \vec{v} \Rightarrow\!\!\!> {\uparrow}Q \dashv \Sigma\,;C$,
(3) Q is C-minimized by $\widehat{\sigma}$, and
(4) $T;\Gamma, x : [\widehat{\sigma}]\,Q \vDash c' : N$.

By the induction hypothesis applied to $T;\Gamma \vDash v\colon {\downarrow}M$, we have $T;\Gamma \vdash v\colon {\downarrow}M$ and $T \vdash {\downarrow}M$ (and thus, $T\,;\mathbf{dom}\,(\Sigma) \vdash M$).

By the induction hypothesis applied to $T;\Gamma, x : [\widehat{\sigma}]\,Q \vDash c' : N$, we have $T \vdash N$ and $T;\Gamma, x : [\widehat{\sigma}]\,Q \vdash c' : N$.

By the induction hypothesis applied to $T\,;\Gamma\,;\cdot \vDash M \bullet \vec{v} \Rightarrow\!\!\!> {\uparrow}Q \dashv \Sigma\,;C$, we have:
(1) $T \vdash^{\supseteq} \Sigma$
(2) $T\,;\mathbf{dom}\,(\Sigma) \vdash {\uparrow}Q$
(3) $\Sigma|_{\mathbf{fav}\, M \cup \mathbf{fav}\, Q} \vdash C$ (and thus, $\mathbf{dom}\,(C) \subseteq \mathbf{fav}\, M \cup \mathbf{fav}\, Q$)
(4) for any $\Sigma \vdash \widehat{\sigma} : C$, we have $T\,;\Gamma \vdash [\widehat{\sigma}] M \bullet \vec{v} \Rightarrow\!\!\!> [\widehat{\sigma}]{\uparrow}Q$, which, since M is ground means $T\,;\Gamma \vdash M \bullet \vec{v} \Rightarrow\!\!\!> {\uparrow}[\widehat{\sigma}]\,Q$.

To infer $T;\Gamma \vdash \mathbf{let}\, x = v(\vec{v})\,;c' : N$, we apply the corresponding declarative rule ($\text{LET}^{\text{INF}}_{@}$). Let us show that the premises hold:

- $T;\Gamma \vdash v\colon {\downarrow}M$ holds by the induction hypothesis;
- $T;\Gamma, x : [\widehat{\sigma}]\,Q \vdash c' : N$ also holds by the induction hypothesis, as noted above;
- $T\,;\Gamma \vdash M \bullet \vec{v} \Rightarrow\!\!\!> {\uparrow}[\widehat{\sigma}]\,Q$ holds, as noted above;
- To show the principality of ${\uparrow}[\widehat{\sigma}]\,Q$, we assume that for some other type R holds $T\,;\Gamma \vdash M \bullet \vec{v} \Rightarrow\!\!\!> {\uparrow}R$, that is $T\,;\Gamma \vdash [\cdot] M \bullet \vec{v} \Rightarrow\!\!\!> {\uparrow}R$. Then by the completeness of typing (lemma 94), there exist N', Σ', and C' such that
  (1) $T\,;\Gamma\,;\cdot \vDash M \bullet \vec{v} \Rightarrow\!\!\!> N' \dashv \Sigma'\,;C'$ and
  (2) there exists a substitution $\Sigma' \vdash \widehat{\sigma}' : C'$ such that $T \vdash [\widehat{\sigma}']\,N' \simeq^{\leqslant} {\uparrow}R$.

  By determinacy of the typing algorithm (lemma 92), $T\,;\Gamma\,;\cdot \vDash M \bullet \vec{v} \Rightarrow\!\!\!> N' \dashv \Sigma'\,;C'$, means that C' is C, Σ' is Σ, and N' is ${\uparrow}Q$. This way, $T \vdash [\widehat{\sigma}']{\uparrow}Q \simeq^{\leqslant} {\uparrow}R$ for substitution $\Sigma \vdash \widehat{\sigma}' : C$. To show the principality, it suffices to notice that $T \vdash R \geqslant [\widehat{\sigma}]\,Q$ or equivalently $T \vdash [\widehat{\sigma}']Q \geqslant [\widehat{\sigma}]Q$, which holds by the soundness of the minimal instantiation (lemma 86) since Q is C-minimized by $\widehat{\sigma}$.

Case 11. ($\text{LET}^{\text{INF}}_{\exists}$) We are proving that if $T;\Gamma \vDash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v; c' : N$ then $T;\Gamma \vdash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v\,; c' : N$ and $T \vdash N$. By the inversion, we have:

(1) $T;\Gamma \vDash v \colon \exists\overrightarrow{\alpha^-}.\, P$
(2) $T, \overrightarrow{\alpha^-}; \Gamma, x : P \vDash c' : N$
(3) $T \vdash N$

By the induction hypothesis applied to $T;\Gamma \vDash v \colon \exists\overrightarrow{\alpha^-}.\, P$, we have $T;\Gamma \vdash v \colon \exists\overrightarrow{\alpha^-}.\, P$ and $\exists\overrightarrow{\alpha^-}.\, P$ is normalized. By the induction hypothesis applied to $T, \overrightarrow{\alpha^-}; \Gamma, x : P \vDash c' : N$, we have $T, \overrightarrow{\alpha^-}; \Gamma, x : P \vdash c' : N$.

To show $T;\Gamma \vdash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v\,; c' : N$, we apply the corresponding declarative rule ($\text{LET}^{\text{INF}}_{\exists}$). Let us show that the premises hold:

(1) $T;\Gamma \vdash v \colon \exists\overrightarrow{\alpha^-}.\, P$ holds by the induction hypothesis, as noted above,
(2) $\mathbf{nf}\,(\exists\overrightarrow{\alpha^-}.\, P) = \exists\overrightarrow{\alpha^-}.\, P$ holds since $\exists\overrightarrow{\alpha^-}.\, P$ is normalized,
(3) $T, \overrightarrow{\alpha^-}; \Gamma, x : P \vdash c' : N$ also holds by the induction hypothesis,
(4) $T \vdash N$ holds by the inversion, as noted above.

Case 12. ($\emptyset^{\text{INF}}_{\bullet\Rrightarrow}$) Then by assumption:

- $T \vdash^{\supseteq} \Sigma$,
- $T\,; \mathbf{dom}\,(\Sigma) \vdash N$ is free from negative algorithmic variables,
- $T\,;\Gamma\,;\Sigma \vDash N \bullet \cdot \Rrightarrow \mathbf{nf}\,(N) \dashv \Sigma\,;\cdot$.

Let us show the required properties:

(1) $T \vdash^{\supseteq} \Sigma$ holds by assumption,
(2) $\Sigma \subseteq \Sigma$ holds trivially,
(3) $\mathbf{nf}\,(N)$ is evidently normalized, $T\,;\mathbf{dom}\,(\Sigma) \vdash N$ implies $T\,;\mathbf{dom}\,(\Sigma) \vdash \mathbf{nf}\,(N)$ by corollary 24, and lemma 38 means that $\mathbf{nf}\,(N)$ is inherently free from negative algorithmic variables,
(4) $\mathbf{dom}\,(\Sigma) \cap \mathbf{fav}(\mathbf{nf}\,(N)) \subseteq \mathbf{fav} N$ holds since $\mathbf{fav}(\mathbf{nf}\,(N)) = \mathbf{fav}(N)$,
(5) $\Sigma|_{\mathbf{fav} N \cup \mathbf{favnf}\,(N)} \vdash \cdot$ holds trivially,
(6) suppose that $\Sigma \vdash \widehat{\sigma} : \mathbf{fav} N \cup \mathbf{favnf}\,(N)$. To show $T\,;\Gamma \vdash [\widehat{\sigma}]N \bullet \cdot \Rrightarrow [\widehat{\sigma}]\mathbf{nf}\,(N)$, we apply the corresponding declarative rule ($\emptyset^{\text{INF}}_{\bullet\Rrightarrow}$). To show $T \vdash [\widehat{\sigma}]N \simeq^{\leqslant} [\widehat{\sigma}]\mathbf{nf}\,(N)$, we apply the following sequence: $N \simeq^{D} \mathbf{nf}\,(N)$ by lemma 39, then $[\widehat{\sigma}]N \simeq^{D} [\widehat{\sigma}]\mathbf{nf}\,(N)$ by corollary 21, then $T \vdash [\widehat{\sigma}]N \simeq^{\leqslant} [\widehat{\sigma}]\mathbf{nf}\,(N)$ by lemma 27.

Case 13. ($\rightarrow^{\text{INF}}_{\bullet\Rrightarrow}$) By assumption:

(1) $T \vdash^{\supseteq} \Sigma$,
(2) $T\,; \mathbf{dom}\,(\Sigma) \vdash Q \rightarrow N$ is free from negative algorithmic variables, and hence, so are Q and N,
(3) $T\,;\Gamma\,;\Sigma \vDash Q \rightarrow N \bullet v, \vec{v} \Rrightarrow M \dashv \Sigma'\,; C$, and by inversion:
  (a) $T;\Gamma \vDash v \colon P$, and by the induction hypothesis applied to this judgment, we have $T;\Gamma \vdash v \colon P$, and $T \vdash P$;
  (b) $T;\, \Sigma \vDash Q \geqslant P \dashv C_1$, and by the soundness of subtyping: $\Sigma \vdash C_1 : \mathbf{fav} Q$ (and thus, $\mathbf{dom}\,(C_1) = \mathbf{fav} Q$), and for any $\Sigma \vdash \widehat{\sigma} : C_1$, we have $T \vdash [\widehat{\sigma}]Q \geqslant P$;
  (c) $T\,;\Gamma\,;\Sigma \vDash N \bullet \vec{v} \Rrightarrow M \dashv \Sigma'\,; C_2$, and by the induction hypothesis applied to this judgment,
    (i) $T \vdash^{\supseteq} \Sigma'$,
    (ii) $\Sigma \subseteq \Sigma'$,
    (iii) $T\,;\mathbf{dom}\,(\Sigma') \vdash M$ is normalized and free from negative algorithmic variables,

(iv) $\mathbf{dom}\,(\Sigma) \cap \mathbf{fav}(M) \subseteq \mathbf{fav}\,N$,
(v) $\Sigma'|_{\mathbf{fav}(M)\cup\mathbf{fav}(N)} \vdash C_2$, and thus, $\mathbf{dom}\,(C_2) \subseteq \mathbf{fav}(M) \cup \mathbf{fav}(N)$,
(vi) for any $\widehat{\sigma}$ such that $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(M) \cup \mathbf{fav}(N)$ and $\Sigma' \vdash \widehat{\sigma} : C_2$, we have $T ; \Gamma \vdash [\widehat{\sigma}]N \bullet \vec{v} \Longrightarrow\!\!\!> [\widehat{\sigma}]M$;

(d) $\Sigma \vdash C_1 \,\&\, C_2 = C$, which by lemma 81 implies $\mathbf{dom}\,(C) = \mathbf{dom}\,(C_1) \cup \mathbf{dom}\,(C_2) \subseteq \mathbf{fav}\,Q \cup \mathbf{fav}\,M \cup \mathbf{fav}\,N$.

Let us show the required properties:

(1) $T \vdash^{\supseteq} \Sigma'$ is shown above,
(2) $\Sigma \subseteq \Sigma'$ is shown above,
(3) $T ; \mathbf{dom}\,(\Sigma') \vdash M$ is normalized and free from negative algorithmic variables, as shown above,
(4) $\mathbf{dom}\,(\Sigma) \cap \mathbf{fav}(M) \subseteq \mathbf{fav}\,N \subseteq \mathbf{fav}(Q \to N)$ (the first inclusion is shown above, the second one is by definition),
(5) To show $\Sigma'|_{\mathbf{fav}(Q)\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)} \vdash C$, first let us notice that $\mathbf{fav}(Q) \cup \mathbf{fav}(N) \cup \mathbf{fav}(M) \subseteq \mathbf{dom}\,(C)$, as mentioned above. Then we demonstrate $\Sigma' \vdash C$: $\Sigma \vdash C_1$ and $\Sigma \subseteq \Sigma'$ imply $\Sigma' \vdash C_1$, by the soundness of constraint merge (lemma 81) applied to $\Sigma' \vdash C_1 \,\&\, C_2 = C$:
   (a) $\Sigma' \vdash C$,
   (b) for any $\Sigma' \vdash \widehat{\sigma} : C$, $\Sigma' \vdash \widehat{\sigma} : C_i$ holds;
(6) Suppose that $\Sigma' \vdash \widehat{\sigma} : \mathbf{fav}(Q) \cup \mathbf{fav}(N) \cup \mathbf{fav}(M)$ and $\Sigma' \vdash \widehat{\sigma} : C$. To show $T ; \Gamma \vdash [\widehat{\sigma}](Q \to N) \bullet v, \vec{v} \Longrightarrow\!\!\!> [\widehat{\sigma}]M$, that is $T ; \Gamma \vdash [\widehat{\sigma}]Q \to [\widehat{\sigma}]N \bullet v, \vec{v} \Longrightarrow\!\!\!> [\widehat{\sigma}]M$, we apply the corresponding declarative rule ($\to^{\mathrm{INF}}_{\bullet\Longrightarrow\!\!\!>}$). Let us show the required premises:
   (a) $T; \Gamma \vdash v \colon P$ holds as shown above,
   (b) $T \vdash [\widehat{\sigma}]Q \geqslant P$ holds since $T \vdash [\widehat{\sigma}|_{\mathbf{fav}(Q)}]Q \geqslant P$ by the soundness of subtyping as noted above: since $\Sigma' \vdash \widehat{\sigma} : C$ implies $\Sigma' \vdash \widehat{\sigma}|_{\mathbf{fav}(Q)} : C_1$, which we strengthen to $\Sigma \vdash \widehat{\sigma}|_{\mathbf{fav}(Q)} : C_1$,
   (c) $T ; \Gamma \vdash [\widehat{\sigma}]N \bullet \vec{v} \Longrightarrow\!\!\!> [\widehat{\sigma}]M$ holds by the induction hypothesis as shown above, since $\Sigma' \vdash \widehat{\sigma} : C$ implies $\Sigma' \vdash \widehat{\sigma} : C_2$, and then $\Sigma' \vdash \widehat{\sigma}|_{\mathbf{fav}(N)\cup\mathbf{fav}(M)} : C_2$ and $\Sigma \vdash \widehat{\sigma}|_{\mathbf{fav}(N)\cup\mathbf{fav}(M)} : \mathbf{fav}(N) \cup \mathbf{fav}(M)$.

Case 14. ($\forall^{\mathrm{INF}}_{\bullet\Longrightarrow\!\!\!>}$)

By assumption:

(1) $T \vdash^{\supseteq} \Sigma$,
(2) $T ; \mathbf{dom}\,(\Sigma) \vdash \forall\overrightarrow{\alpha^+}.\, N$ is free from negative algorithmic variables,
(3) $T ; \Gamma ; \Sigma \vDash \forall\overrightarrow{\alpha^+}.\, N \bullet \vec{v} \Longrightarrow\!\!\!> M \dashv \Sigma' ; C$, which by inversion means $\vec{v} \neq \cdot$, $\overrightarrow{\alpha^+} \neq \cdot$, and $T ; \Gamma ; \Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vDash [\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]N \bullet \vec{v} \Longrightarrow\!\!\!> M \dashv \Sigma' ; C$. It is easy to see that the induction hypothesis is applicable to the latter judgment:
   - $T \vdash^{\supseteq} \Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\}$ holds by $T \vdash^{\supseteq} \Sigma$,
   - $T ; \mathbf{dom}\,(\Sigma), \overrightarrow{\widehat{\alpha}^+} \vdash [\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]N$ holds since $T ; \mathbf{dom}\,(\Sigma) \vdash \forall\overrightarrow{\alpha^+}.\, N$ $[\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]N$ is normalized and free from negative algorithmic variables since so is N;

   This way, by the inductive hypothesis applied to $T ; \Gamma ; \Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vDash [\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]N \bullet \vec{v} \Longrightarrow\!\!\!> M \dashv \Sigma' ; C$, we have:
   (a) $T \vdash^{\supseteq} \Sigma'$,
   (b) $\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \subseteq \Sigma'$,
   (c) $T ; \mathbf{dom}\,(\Sigma') \vdash M$ is normalized and free from negative algorithmic variables,
   (d) $\mathbf{dom}\,(\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\}) \cap \mathbf{fav}(M) \subseteq \mathbf{fav}([\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]N)$,

(e) $\Sigma'|_{\Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)} \vdash C$, where Υ denotes $\mathbf{fav}([\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N)\cap\overrightarrow{\widehat{\alpha}^{+}}$, that is the algorithmization of the $\forall$-variables that are actually used in N.

(f) for any $\widehat{\sigma}$ such that $\Sigma' \vdash \widehat{\sigma} : \Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)$ and $\Sigma' \vdash \widehat{\sigma} : C$, we have $T\,;\Gamma \vdash [\widehat{\sigma}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N \bullet \vec{v} \Rrightarrow [\widehat{\sigma}]M$.

Let us show the required properties:

(1) $T \vdash^{\supseteq} \Sigma'$ is shown above;

(2) $\Sigma \subseteq \Sigma'$ since $\Sigma, \overrightarrow{\widehat{\alpha}^{+}}\{T\} \subseteq \Sigma'$;

(3) $T\,;\mathbf{dom}\,(\Sigma') \vdash M$ is normalized and free from negative algorithmic variables, as shown above;

(4) $\mathbf{dom}\,(\Sigma)\cap\mathbf{fav}(M) \subseteq \mathbf{fav}(N)$ since $\mathbf{dom}\,(\Sigma, \overrightarrow{\widehat{\alpha}^{+}}\{T\})\cap\mathbf{fav}(M) \subseteq \mathbf{fav}([\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N)$ implies $(\mathbf{dom}\,(\Sigma)\cup\overrightarrow{\widehat{\alpha}^{+}})\cap\mathbf{fav}(M) \subseteq \mathbf{fav}(N)\cup\overrightarrow{\widehat{\alpha}^{+}}$, thus, $\mathbf{dom}\,(\Sigma)\cap\mathbf{fav}(M) \subseteq \mathbf{fav}(N)\cup\overrightarrow{\widehat{\alpha}^{+}}$, and since $\mathbf{dom}\,(\Sigma)$ is disjoint with $\overrightarrow{\widehat{\alpha}^{+}}$, $\mathbf{dom}\,(\Sigma)\cap\mathbf{fav}(M) \subseteq \mathbf{fav}(N)$;

(5) $\Sigma'|_{\mathbf{fav}(N)\cup\mathbf{fav}(M)} \vdash C|_{\mathbf{fav}(N)\cup\mathbf{fav}(M)}$ follows from $\Sigma'|_{\Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)} \vdash C$ if we restrict both sides to $\mathbf{fav}(N)\cup\mathbf{fav}(M)$.

(6) Let us assume $\Sigma' \vdash \widehat{\sigma} : \mathbf{fav}(N)\cup\mathbf{fav}(M)$ and $\Sigma' \vdash \widehat{\sigma} : C|_{\mathbf{fav}(N)\cup\mathbf{fav}(M)}$. Then to show $T\,;\Gamma \vdash [\widehat{\sigma}]\forall\overrightarrow{\alpha^{+}}.\,N \bullet \vec{v} \Rrightarrow [\widehat{\sigma}]M$, that is $T\,;\Gamma \vdash \forall\overrightarrow{\alpha^{+}}.\,[\widehat{\sigma}]N \bullet \vec{v} \Rrightarrow [\widehat{\sigma}]M$, we apply the corresponding declarative rule ($\forall^{\mathrm{INF}}_{\bullet\Rrightarrow}$). To do so, we need to provide a substitution for $\overrightarrow{\alpha^{+}}$, i.e. $T \vdash \sigma_0 : \overrightarrow{\alpha^{+}}$ such that $T\,;\Gamma \vdash [\sigma_0][\widehat{\sigma}]N \bullet \vec{v} \Rrightarrow [\widehat{\sigma}]M$.

By lemma 76, we construct $\widehat{\sigma}_0$ such that $\Sigma' \vdash \widehat{\sigma}_0 : \overrightarrow{\widehat{\alpha}^{+}}$ and $\Sigma' \vdash \widehat{\sigma}_0 : C|_{\overrightarrow{\widehat{\alpha}^{+}}}$.

Then σ_0 is defined as $\widehat{\sigma}_0 \circ \widehat{\sigma}|_{\overrightarrow{\widehat{\alpha}^{+}}} \circ \overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}$.

Let us show that the premises of ($\forall^{\mathrm{INF}}_{\bullet\Rrightarrow}$) hold:

- To show $T \vdash \sigma_0 : \overrightarrow{\alpha^{+}}$, let us take $\alpha_i^{+} \in \overrightarrow{\alpha^{+}}$. If $\widehat{\alpha}_i^{+} \in \mathbf{fav}(M)$ then $[\sigma_0]\alpha_i^{+} = [\widehat{\sigma}]\widehat{\alpha}_i^{+}$, and $\Sigma' \vdash \widehat{\sigma} : \mathbf{fav}(N)\cup\mathbf{fav}(M)$ implies $\Sigma'(\widehat{\alpha}_i^{+}) \vdash [\widehat{\sigma}]\widehat{\alpha}^{+}$. Analogously, if $\widehat{\alpha}_i^{+} \in \overrightarrow{\widehat{\alpha}^{+}} \setminus \mathbf{fav}(M)$ then $[\sigma_0]\alpha_i^{+} = [\widehat{\sigma}_0]\widehat{\alpha}_i^{+}$, and $\Sigma' \vdash \widehat{\sigma}_0 : \overrightarrow{\widehat{\alpha}^{+}}$ implies $\Sigma'(\widehat{\alpha}_i^{+}) \vdash [\widehat{\sigma}_0]\widehat{\alpha}_i^{+}$. In any case, $\Sigma'(\widehat{\alpha}_i^{+}) \vdash [\sigma]\alpha_i^{+}$ can be weakened to $T \vdash [\sigma_0]\alpha_i^{+}$, since $T \vdash^{\supseteq} \Sigma'$.
- Let us show $T\,;\Gamma \vdash [\sigma_0][\widehat{\sigma}]N \bullet \vec{v} \Rrightarrow [\widehat{\sigma}]M$. It suffices to construct $\widehat{\sigma}_1$ such that
  (a) $\Sigma' \vdash \widehat{\sigma}_1 : \Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)$,
  (b) $\Sigma' \vdash \widehat{\sigma}_1 : C$,
  (c) $[\sigma_0][\widehat{\sigma}]N = [\widehat{\sigma}_1][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N$, and
  (d) $[\widehat{\sigma}]M = [\widehat{\sigma}_1]M$,

  because then we can apply the induction hypothesis (3f) to $\widehat{\sigma}_1$, rewrite the conclusion by $[\widehat{\sigma}_1][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N = [\sigma_0][\widehat{\sigma}]N$ and $[\widehat{\sigma}_1]M = [\widehat{\sigma}]M$, and infer the required judgement.

  Let us take $\widehat{\sigma}_1 = (\widehat{\sigma}_0 \circ \widehat{\sigma})|_{\Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)}$, then
  (a) $\Sigma' \vdash \widehat{\sigma}_1 : \Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)$, since $\Sigma' \vdash \widehat{\sigma}_0 : \overrightarrow{\widehat{\alpha}^{+}}$ and $\Sigma' \vdash \widehat{\sigma} : \mathbf{fav}(N)\cup\mathbf{fav}(M)$, we have $\Sigma' \vdash \widehat{\sigma}_0 \circ \widehat{\sigma} : \overrightarrow{\widehat{\alpha}^{+}}\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)$, which we restrict to $\Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)$.
  (b) $\Sigma' \vdash \widehat{\sigma}_1 : C$, Let us take any constraint $e \in C$ restricting variable $\widehat{\beta}^{\pm}$. $\Sigma'|_{\Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)} \vdash C$ implies that $\widehat{\beta}^{\pm} \in \Upsilon\cup\mathbf{fav}(N)\cup\mathbf{fav}(M)$.
  If $\widehat{\beta}^{\pm} \in \mathbf{fav}(N)\cup\mathbf{fav}(M)$ then $[\widehat{\sigma}_1]\widehat{\beta}^{\pm} = [\widehat{\sigma}]\widehat{\beta}^{\pm}$. Additionally, $e \in C|_{\mathbf{fav}(N)\cup\mathbf{fav}(M)}$, which, since $\Sigma' \vdash \widehat{\sigma} : C|_{\mathbf{fav}(N)\cup\mathbf{fav}(M)}$, means $\Sigma'(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}]\widehat{\beta}^{\pm} : e$.

If $\widehat{\beta}^{\pm} \in \Upsilon \setminus (\mathbf{fav}(N) \cup \mathbf{fav}(M))$ then $[\widehat{\sigma}_1]\widehat{\beta}^{\pm} = [\widehat{\sigma}_0]\widehat{\beta}^{\pm}$. Additionally, $e \in C|_{\overrightarrow{\widehat{\alpha}^{+}}}$, which, since $\Sigma' \vdash \widehat{\sigma}_0 : C|_{\overrightarrow{\widehat{\alpha}^{+}}}$, means $\Sigma'(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}_0]\widehat{\beta}^{\pm} : e$.

(c) Let us prove $[\sigma_0][\widehat{\sigma}]N = [\widehat{\sigma}_1][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N$ by the following reasoning

$$\begin{aligned}
[\sigma_0][\widehat{\sigma}]N &= [\widehat{\sigma}_0][\widehat{\sigma}|_{\overrightarrow{\widehat{\alpha}^{+}}}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}][\widehat{\sigma}]N && \text{by definition of } \sigma_0\\
&= [\widehat{\sigma}_0][\widehat{\sigma}|_{\overrightarrow{\widehat{\alpha}^{+}}}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}][\widehat{\sigma}|_{\mathbf{fav}(N)}]N && \text{by lemma 55}\\
&= [\widehat{\sigma}_0][\widehat{\sigma}|_{\overrightarrow{\widehat{\alpha}^{+}}}][\widehat{\sigma}|_{\mathbf{fav}(N)}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N && \mathbf{fav}(N) \cap \overrightarrow{\widehat{\alpha}^{+}} = \emptyset \text{ and } \overrightarrow{\alpha^{+}} \cap T = \emptyset\\
&= [\widehat{\sigma}|_{\overrightarrow{\widehat{\alpha}^{+}}}][\widehat{\sigma}|_{\mathbf{fav}(N)}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N && [\widehat{\sigma}|_{\overrightarrow{\widehat{\alpha}^{+}}}][\widehat{\sigma}|_{\mathbf{fav}(N)}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N \text{ is ground}\\
&= [\widehat{\sigma}|_{\overrightarrow{\widehat{\alpha}^{+}} \cup \mathbf{fav}(N)}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N\\
&= [\widehat{\sigma}|_{\Upsilon \cup \mathbf{fav}(N)}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N && \text{by lemma 55: } \mathbf{fav}([\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N) = \Upsilon \cup \mathbf{fav}(N)\\
&= [\widehat{\sigma}|_{\Upsilon \cup \mathbf{fav}(N) \cup \mathbf{fav}(M)}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N && \text{also by lemma 55}\\
&= [(\widehat{\sigma}_0 \circ \widehat{\sigma})|_{\Upsilon \cup \mathbf{fav}(N) \cup \mathbf{fav}(M)}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N && [\widehat{\sigma}|_{\Upsilon \cup \mathbf{fav}(N) \cup \mathbf{fav}(M)}][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N \text{ is ground}\\
&= [\widehat{\sigma}_1][\overrightarrow{\widehat{\alpha}^{+}}/\overrightarrow{\alpha^{+}}]N && \text{by definition of } \widehat{\sigma}_1
\end{aligned}$$

(d) $[\widehat{\sigma}]M = [\widehat{\sigma}_1]M$ By definition of $\widehat{\sigma}_1$, $[\widehat{\sigma}_1]M$ is equal to $[(\widehat{\sigma}_0 \circ \widehat{\sigma})|_{\Upsilon \cup \mathbf{fav}(N) \cup \mathbf{fav}(M)}]M$, which by lemma 55 is equal to $[\widehat{\sigma}_0 \circ \widehat{\sigma}]M$, that is $[\widehat{\sigma}_0][\widehat{\sigma}]M$, and since $[\widehat{\sigma}]M$ is ground, $[\widehat{\sigma}_0][\widehat{\sigma}]M = [\widehat{\sigma}]M$.

- $\overrightarrow{\alpha^{+}} \neq \cdot$ and $\overrightarrow{v} \neq \cdot$ hold by assumption.

□

Lemma 94 (Completeness of Typing). *Suppose that $T \vdash \Gamma$. For an inference tree T_1,*

$+$ *If T_1 infers $T;\Gamma \vdash v\colon P$ then $T;\Gamma \vDash v\colon \mathbf{nf}\,(P)$*

$-$ *If T_1 infers $T;\Gamma \vdash c\colon N$ then $T;\Gamma \vDash c\colon \mathbf{nf}\,(N)$*

- *If T_1 infers $T\,;\Gamma \vdash [\widehat{\sigma}]N \bullet \overrightarrow{v} \Longrightarrow\!\!\!> M$ and*
  (1) $T \vdash^{\supseteq} \Sigma$,
  (2) $T \vdash M$,
  (3) $T\,;\mathbf{dom}\,(\Sigma) \vdash N$ (free from negative algorithmic variables, that is $\widehat{\alpha}^{-} \notin \mathbf{fav}\,N$), and
  (4) $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N)$,
  then there exist M', Σ', and C such that
  (1) $T\,;\Gamma\,;\Sigma \vDash N \bullet \overrightarrow{v} \Longrightarrow\!\!\!> M' \dashv \Sigma'\,;C$ and
  (2) for any $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N)$ and $T \vdash M$ such that $T\,;\Gamma \vdash [\widehat{\sigma}]N \bullet \overrightarrow{v} \Longrightarrow\!\!\!> M$, there exists $\widehat{\sigma}'$ such that
    (a) $\Sigma' \vdash \widehat{\sigma}' : \mathbf{fav}\,N \cup \mathbf{fav}\,M'$ and $\Sigma' \vdash \widehat{\sigma}' : C$,
    (b) $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}\,N$, and
    (c) $T \vdash [\widehat{\sigma}']M' \simeq^{\leqslant} M$.

Proof. We prove it by induction on $\mathrm{metric}(T_1)$, mutually with the soundness of typing (lemma 93). Let us consider the last rule applied to infer the derivation.

Case 1. $(\{\}^{\text{INF}})$
Then we are proving that if $T;\Gamma \vdash \{c\}\colon \downarrow N$ (inferred by $(\{\}^{\text{INF}})$) then $T;\Gamma \vDash \{c\}\colon \mathbf{nf}\,(\downarrow N)$. By inversion of $T;\Gamma \vdash \{c\}\colon \downarrow N$, we have $T;\Gamma \vdash c\colon N$, which we apply the induction hypothesis to to obtain $T;\Gamma \vDash c\colon \mathbf{nf}\,(N)$. Then by $(\{\}^{\text{INF}})$, we have $T;\Gamma \vDash \{c\}\colon \downarrow\mathbf{nf}\,(N)$. It is left to notice that $\downarrow\mathbf{nf}\,(N) = \mathbf{nf}\,(\downarrow N)$.

Case 2. ($\textsc{ret}^{\textsc{inf}}$)

The proof is symmetric to the previous case (case 1).

Case 3. ($\textsc{ann}_+^{\textsc{inf}}$)

Then we are proving that if $T; \Gamma \vdash (v : Q) \colon Q$ is inferred by ($\textsc{ann}_+^{\textsc{inf}}$) then $T; \Gamma \vDash (v : Q) \colon \mathbf{nf}\,(Q)$. By inversion, we have:

(1) $T \vdash Q$;
(2) $T; \Gamma \vdash v \colon P$, which by the induction hypothesis implies $T; \Gamma \vDash v \colon \mathbf{nf}\,(P)$;
(3) $T \vdash Q \geqslant P$, and by transitivity, $T \vdash Q \geqslant \mathbf{nf}\,(P)$; Since Q is ground, we have $T\,; \cdot \vdash \boxed{Q}$ and $T \vdash [\cdot]\boxed{Q} \geqslant \mathbf{nf}\,(P)$. Then by the completeness of subtyping (lemma 79), we have $T; \cdot \vDash \boxed{Q} \geqslant \mathbf{nf}\,(P) \dashv C$, where $\cdot \vdash C$ (implying $C = \cdot$). This way, $T; \cdot \vDash \boxed{Q} \geqslant \mathbf{nf}\,(P) \dashv \cdot$.

Then we can apply ($\textsc{ann}_+^{\textsc{inf}}$) to $T \vdash Q$, $T; \Gamma \vDash v \colon \mathbf{nf}\,(P)$ and $T; \cdot \vDash \boxed{Q} \geqslant \mathbf{nf}\,(P) \dashv \cdot$ to infer $T; \Gamma \vDash (v : Q) \colon \mathbf{nf}\,(Q)$.

Case 4. ($\textsc{ann}_-^{\textsc{inf}}$)

The proof is symmetric to the previous case (case 3).

Case 5. ($\lambda^{\textsc{inf}}$)

Then we are proving that if $T; \Gamma \vdash \lambda x : P.\ c \colon P \to N$ is inferred by ($\lambda^{\textsc{inf}}$), then $T; \Gamma \vDash \lambda x : P.\ c \colon \mathbf{nf}\,(P \to N)$.

By inversion of $T; \Gamma \vdash \lambda x : P.\ c \colon P \to N$, we have $T \vdash P$ and $T; \Gamma, x : P \vdash c \colon N$. Then by the induction hypothesis, $T; \Gamma, x : P \vDash c \colon \mathbf{nf}\,(N)$. By ($\lambda^{\textsc{inf}}$), we infer $T; \Gamma \vDash \lambda x : P.\ c \colon \mathbf{nf}\,(P \to \mathbf{nf}\,(N))$. By idempotence of normalization (lemma 44), $\mathbf{nf}\,(P \to \mathbf{nf}\,(N)) = \mathbf{nf}\,(P \to N)$, which concludes the proof for this case.

Case 6. ($\Lambda^{\textsc{inf}}$)

Then we are proving that if $T; \Gamma \vdash \Lambda\alpha^+.\ c \colon \forall\alpha^+.\ N$ is inferred by ($\Lambda^{\textsc{inf}}$), then $T; \Gamma \vDash \Lambda\alpha^+.\ c \colon \mathbf{nf}\,(\forall\alpha^+.\ N)$. Similar to the previous case, by inversion of $T; \Gamma \vdash \Lambda\alpha^+.\ c \colon \forall\alpha^+.\ N$, we have $T, \alpha^+; \Gamma \vdash c \colon N$, and then by the induction hypothesis, $T, \alpha^+; \Gamma \vDash c \colon \mathbf{nf}\,(N)$. After that, application of ($\Lambda^{\textsc{inf}}$), gives as $T; \Gamma \vDash \Lambda\alpha^+.\ c \colon \mathbf{nf}\,(\forall\alpha^+.\ \mathbf{nf}\,(N))$.

It is left to show that $\mathbf{nf}\,(\forall\alpha^+.\ \mathbf{nf}\,(N)) = \mathbf{nf}\,(\forall\alpha^+.\ N)$. Assume $N = \forall\overrightarrow{\beta^+}.\ M$ (where M does not start with $\forall$).

- Then by definition, $\mathbf{nf}\,(\forall\alpha^+.\ N) = \mathbf{nf}\,(\forall\alpha^+, \overrightarrow{\beta^+}.\ M) = \forall\overrightarrow{\gamma^+}.\ \mathbf{nf}\,(M)$, where $\mathbf{ord}\ \alpha^+, \overrightarrow{\beta^+}\ \mathbf{in}\ \mathbf{nf}\,(M) = \overrightarrow{\gamma^+}$.
- On the other hand, $\mathbf{nf}\,(N) = \forall\overrightarrow{\gamma^+}'.\ \mathbf{nf}\,(M)$, where $\mathbf{ord}\ \overrightarrow{\beta^+}\ \mathbf{in}\ \mathbf{nf}\,(M) = \overrightarrow{\gamma^+}'$, and thus, $\mathbf{nf}\,(\forall\alpha^+.\ \mathbf{nf}\,(N)) = \mathbf{nf}\,(\forall\alpha^+, \overrightarrow{\gamma^+}'.\ \mathbf{nf}\,(M)) = \forall\overrightarrow{\gamma^+}''.\ \mathbf{nf}\,(\mathbf{nf}\,(M)) = \forall\overrightarrow{\gamma^+}''.\ \mathbf{nf}\,(M)$, where $\mathbf{ord}\ \alpha^+, \overrightarrow{\gamma^+}'\ \mathbf{in}\ \mathbf{nf}\,(\mathbf{nf}\,(M)) = \overrightarrow{\gamma^+}''$.

It is left to show that $\overrightarrow{\gamma^+}'' = \overrightarrow{\gamma^+}$.

$$\begin{aligned}
\overrightarrow{\gamma^+}'' &= \mathbf{ord}\ \alpha^+, \overrightarrow{\gamma^+}'\ \mathbf{in}\ \mathbf{nf}\,(\mathbf{nf}\,(M)) \\
&= \mathbf{ord}\ \alpha^+, \overrightarrow{\gamma^+}'\ \mathbf{in}\ \mathbf{nf}\,(M) && \text{by idempotence (lemma 44)} \\
&= \mathbf{ord}\ \alpha^+ \cup \overrightarrow{\beta^+} \cap \mathbf{fv}\,\mathbf{nf}\,(M)\ \mathbf{in}\ \mathbf{nf}\,(M) && \text{by definition of } \overrightarrow{\gamma^+}' \text{ and lemma 33} \\
&= \mathbf{ord}\ (\alpha^+ \cup \overrightarrow{\beta^+} \cap \mathbf{fv}\,\mathbf{nf}\,(M)) \cap \mathbf{fv}\,\mathbf{nf}\,(M)\ \mathbf{in}\ \mathbf{nf}\,(M) && \text{by lemma 34} \\
&= \mathbf{ord}\ (\alpha^+ \cup \overrightarrow{\beta^+}) \cap \mathbf{fv}\,\mathbf{nf}\,(M)\ \mathbf{in}\ \mathbf{nf}\,(M) && \text{by set properties} \\
&= \mathbf{ord}\ \alpha^+, \overrightarrow{\beta^+}\ \mathbf{in}\ \mathbf{nf}\,(M) \\
&= \overrightarrow{\gamma^+}
\end{aligned}$$

Case 7. ($\textsc{let}^{\text{INF}}_{\exists}$)
Then we are proving that if $T;\Gamma \vdash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v\,; c\colon N$ is inferred by ($\textsc{let}^{\text{INF}}_{\exists}$), then $T;\Gamma \vDash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v\,; c\colon \mathbf{nf}\,(N)$.
By inversion of $T;\Gamma \vdash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v\,; c\colon N$, we have
(1) $\mathbf{nf}\,(\exists\overrightarrow{\alpha^-}.\,P) = \exists\overrightarrow{\alpha^-}.\,P$,
(2) $T;\Gamma \vdash v\colon \exists\overrightarrow{\alpha^-}.\,P$, which by the induction hypothesis implies $T;\Gamma \vDash v\colon \mathbf{nf}\,(\exists\overrightarrow{\alpha^-}.\,P)$, and hence, $T;\Gamma \vDash v\colon \exists\overrightarrow{\alpha^-}.\,P$.
(3) $T, \overrightarrow{\alpha^-}; \Gamma, x : P \vdash c\colon N$, and by the induction hypothesis, $T, \overrightarrow{\alpha^-}; \Gamma, x : P \vDash c\colon \mathbf{nf}\,(N)$.
(4) $T \vdash N$.
This way, we can apply ($\textsc{let}^{\text{INF}}_{\exists}$) to infer $T;\Gamma \vDash \mathbf{let}^{\exists}(\overrightarrow{\alpha^-}, x) = v\,; c\colon \mathbf{nf}\,(N)$.

Case 8. ($\simeq^{\text{INF}}_{+}$)
Then we are proving that if $T;\Gamma \vdash v\colon P'$ is inferred by ($\simeq^{\text{INF}}_{+}$), then $T;\Gamma \vDash v\colon \mathbf{nf}\,(P')$. By inversion, $T;\Gamma \vdash v\colon P$ and $T \vdash P \simeq^{\leqslant} P'$, and the metric of the tree inferring $T;\Gamma \vdash v\colon P$ is less than the one inferring $T;\Gamma \vdash v\colon P'$. Then by the induction hypothesis, $T;\Gamma \vDash v\colon \mathbf{nf}\,(P)$. By lemma 46 $T \vdash P \simeq^{\leqslant} P'$ implies $\mathbf{nf}\,(P) = \mathbf{nf}\,(P')$, and thus, $T;\Gamma \vDash v\colon \mathbf{nf}\,(P)$ can be rewritten to $T;\Gamma \vDash v\colon \mathbf{nf}\,(P')$.

Case 9. ($\textsc{Var}^{\text{INF}}$)
Then we are proving that $T;\Gamma \vdash x\colon P$ implies $T;\Gamma \vDash x\colon \mathbf{nf}\,(P)$. By inversion of $T;\Gamma \vdash x\colon P$, we have $x : P \in \Gamma$. Then ($\textsc{Var}^{\text{INF}}$) applies to infer $T;\Gamma \vDash x\colon \mathbf{nf}\,(P)$.

Case 10. ($\textsc{let}^{\text{INF}}$)
Then we are proving that $T;\Gamma \vdash \mathbf{let}\,x = v(\overrightarrow{v})\,; c\colon N$ implies $T;\Gamma \vDash \mathbf{let}\,x = v(\overrightarrow{v})\,; c\colon \mathbf{nf}\,(N)$.
By inversion of $T;\Gamma \vdash \mathbf{let}\,x = v(\overrightarrow{v})\,; c\colon N$, we have
(1) $T;\Gamma \vdash v\colon P$, and by the induction hypothesis, $T;\Gamma \vDash v\colon \mathbf{nf}\,(P)$.
(2) $T;\Gamma, x : P \vdash c\colon N$, and by lemma 48, since $T \vdash P \simeq^{\leqslant} \mathbf{nf}\,(P)$, we have $T;\Gamma, x : \mathbf{nf}\,(P) \vdash c\colon N$. Then by the induction hypothesis, $T;\Gamma, x : \mathbf{nf}\,(P) \vDash c\colon \mathbf{nf}\,(N)$.
Together, $T;\Gamma \vDash v\colon \mathbf{nf}\,(P)$ and $T;\Gamma, x : \mathbf{nf}\,(P) \vDash c\colon \mathbf{nf}\,(N)$ imply $T;\Gamma \vDash \mathbf{let}\,x = v(\overrightarrow{v})\,; c\colon \mathbf{nf}\,(N)$ by ($\textsc{let}^{\text{INF}}$).

Case 11. ($\textsc{let}^{\text{INF}}_{:@}$)
Then we prove that $T;\Gamma \vdash \mathbf{let}\,x : P = v(\overrightarrow{v})\,; c\colon N$ implies $T;\Gamma \vDash \mathbf{let}\,x : P = v(\overrightarrow{v})\,; c\colon \mathbf{nf}\,(N)$.
By inversion of $T;\Gamma \vdash \mathbf{let}\,x : P = v(\overrightarrow{v})\,; c\colon N$, we have
(1) $T \vdash P$
(2) $T;\Gamma \vdash v\colon {\downarrow}M$ for some ground M, which by the induction hypothesis means $T;\Gamma \vDash v\colon {\downarrow}\mathbf{nf}\,(M)$
(3) $T\,;\Gamma \vdash M \bullet \overrightarrow{v} \implies {\uparrow}Q$. By lemma 49, since $T \vdash M \simeq^{\leqslant} \mathbf{nf}\,(M)$, we have $T\,;\Gamma \vdash [\cdot]\mathbf{nf}\,(M) \bullet \overrightarrow{v} \implies {\uparrow}Q$, which by the induction hypothesis means that there exist normalized M', Σ, and C_1 such that (noting that M is ground):
  (a) $T\,;\Gamma\,;\cdot \vDash \mathbf{nf}\,(M) \bullet \overrightarrow{v} \implies M' \dashv \Sigma\,; C_1$, where by the soundness, $T\,;\mathbf{dom}\,(\Sigma) \vdash M'$ and $\Sigma \vdash C_1$.
  (b) for any $T \vdash M''$ such that $T\,;\Gamma \vdash \mathbf{nf}\,(M) \bullet \overrightarrow{v} \implies M''$ there exists $\widehat{\sigma}$ such that
    (i) $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\,M'$, $\Sigma \vdash \widehat{\sigma} : C_1$, and
    (ii) $T \vdash [\widehat{\sigma}]M' \simeq^{\leqslant} M''$,
  In particular, there exists $\widehat{\sigma}_0$ such that $\Sigma \vdash \widehat{\sigma}_0 : \mathbf{fav}\,M'$, $\Sigma \vdash \widehat{\sigma}_0 : C_1$, $T \vdash [\widehat{\sigma}_0]M' \simeq^{\leqslant} {\uparrow}Q$. Since M' is normalized and free of negative algorithmic variables, the latter equivalence means $M' = {\uparrow}Q_0$ for some Q_0, and $T \vdash [\widehat{\sigma}_0]Q_0 \simeq^{\leqslant} Q$.
(4) $T \vdash {\uparrow}Q \leqslant {\uparrow}P$, and by transitivity, since $T \vdash [\widehat{\sigma}_0]{\uparrow}Q_0 \simeq^{\leqslant} {\uparrow}Q$, we have $T \vdash [\widehat{\sigma}_0]{\uparrow}Q_0 \leqslant {\uparrow}P$. Let us apply lemma 85 to $T \vdash [\widehat{\sigma}_0]{\uparrow}Q_0 \leqslant {\uparrow}P$ and obtain $\Sigma \vdash C_2$ such that
  (a) $T;\,\Sigma \vDash {\uparrow}Q_0 \leqslant {\uparrow}P \dashv C_2$ and

(b) $\Sigma \vdash \widehat{\sigma}_0 : C_2$.
(5) $T; \Gamma, x : P \vdash c \colon N$, and by the induction hypothesis, $T; \Gamma, x : P \vDash c \colon \mathbf{nf}\,(N)$.

To infer $T; \Gamma \vDash \mathbf{let}\, x : P = v(\vec{v})\,;\, c \colon \mathbf{nf}\,(N)$, we apply the corresponding algorithmic rule ($\text{LET}^{\text{INF}}_{:@}$). Let us show that the premises hold:

(1) $T \vdash P$,
(2) $T; \Gamma \vDash v \colon \downarrow\mathbf{nf}\,(M)$,
(3) $T\,;\Gamma\,;\cdot \vDash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow \uparrow Q_0 \dashv \Sigma\,; C_1$,
(4) $T;\, \Sigma \vDash \uparrow Q_0 \leqslant \uparrow P \dashv C_2$, and
(5) $T; \Gamma, x : P \vDash c \colon \mathbf{nf}\,(N)$ hold as noted above;
(6) $\Sigma \vdash C_1 \,\&\, C_2 = C$ is defined by lemma 83, since $\Sigma \vdash \widehat{\sigma}_0 : C_1$ and $\Sigma \vdash \widehat{\sigma}_0 : C_2$.

Case 12. ($\text{LET}^{\text{INF}}_{@}$)

By assumption, c is $\mathbf{let}\, x = v(\vec{v})\,;\, c'$. Then by inversion of $T; \Gamma \vdash \mathbf{let}\, x = v(\vec{v})\,;\, c' \colon N$:

- $T; \Gamma \vdash v \colon \downarrow M$, which by the induction hypothesis means $T; \Gamma \vDash v \colon \downarrow\mathbf{nf}\,(M)$;
- $T\,; \Gamma \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q$ principal. Then by lemma 49, since $T \vdash M \simeq^{\leqslant} \mathbf{nf}\,(M)$, we have $T\,;\Gamma \vdash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow \uparrow Q$ and moreover, $T\,;\Gamma \vdash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow \uparrow Q$ principal: since for any inference, $\mathbf{nf}\,(M)$ can be replaced back with M, the sets of types Q' inferred for the applications $T\,;\Gamma \vdash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow \uparrow Q'$ and $T\,;\Gamma \vdash M \bullet \vec{v} \Rrightarrow \uparrow Q'$ are the same. Then the induction hypothesis applied to $T\,;\Gamma \vdash [\cdot]\mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow \uparrow Q$ implies that there exist M', Σ, and C such that (considering M is ground):
  (1) $T\,;\Gamma\,;\cdot \vDash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow M' \dashv \Sigma\,; C$, which, by the soundness, implies, in particular that
    (a) $T\,; \mathbf{dom}\,(\Sigma) \vdash M'$ is normalized and free of negative algorithmic variables,
    (b) $\Sigma|_{\mathbf{fav}(M')} \vdash C$, which means $\mathbf{dom}\,(C) \subseteq \mathbf{fav}(M')$,
    (c) for any $\Sigma \vdash \widehat{\sigma} : \mathbf{fav} M'$ such that $\Sigma \vdash \widehat{\sigma} : C$, we have $T\,;\Gamma \vdash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow [\widehat{\sigma}]M'$.

  and
  (2) for any $T \vdash M''$ such that $T\,;\Gamma \vdash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow M''$, (and in particular, for $T \vdash \uparrow Q$) there exists $\widehat{\sigma}_1$ such that
    (a) $\Sigma \vdash \widehat{\sigma}_1 : \mathbf{fav} M'$, $\Sigma \vdash \widehat{\sigma}_1 : C$, and
    (b) $T \vdash [\widehat{\sigma}_1]M' \simeq^{\leqslant} M''$, and in particular, $T \vdash [\widehat{\sigma}_1]M' \simeq^{\leqslant} \uparrow Q$. Since M' is normalized and free of negative algorithmic variables, it means that $M' = \uparrow P$ for some P ($T\,; \mathbf{dom}\,(\Sigma) \vdash P$) that is $T \vdash [\widehat{\sigma}_1]P \simeq^{\leqslant} Q$.
- $T; \Gamma, x : Q \vdash c' \colon N$

To infer $T; \Gamma \vDash \mathbf{let}\, x = v(\vec{v})\,;\, c' \colon \mathbf{nf}\,(N)$, let us apply the corresponding algorithmic rule (($\text{LET}^{\text{INF}}_{@}$)):

(1) $T; \Gamma \vDash v \colon \downarrow\mathbf{nf}\,(M)$ holds as noted above;
(2) $T\,;\Gamma\,;\cdot \vDash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow \uparrow P \dashv \Sigma\,; C$ holds as noted above;
(3) Let us show that $nf(iQ)$ is the minimal instantiation of P w.r.t. C, in other words, P is C-minimized by $\widehat{\sigma}$ for some $\widehat{\sigma}$ and $[\widehat{\sigma}]P = \mathbf{nf}\,(Q)$. By rewriting $\mathbf{nf}\,(Q)$ as $\mathbf{nf}\,([\widehat{\sigma}_1]P)$, we need to show $[\widehat{\sigma}]P = \mathbf{nf}\,([\widehat{\sigma}_1]P)$.
    Let us apply the completeness of minimal instantiation (lemma 87). That would give us $\widehat{\sigma} = \mathbf{nf}\,(\widehat{\sigma}_1)$, which would immediately imply the required equality. To do that, we need to demonstrate that $\widehat{\sigma}_1$ is the minimal instantiation of P w.r.t. C. In other words, any other substitution respecting C, instantiate P into a *supertype* of Q. To do that, we apply the principality of Q: $T\,;\Gamma \vdash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow \uparrow Q$ principal: which means that for any other Q' such that $T\,;\Gamma \vdash \mathbf{nf}\,(M) \bullet \vec{v} \Rrightarrow \uparrow Q'$, we have $T \vdash Q' \geqslant Q$. It is left

to show that any substitution respecting C gives us Q' inferrable for the application $T\,;\Gamma \vdash M \bullet \overrightarrow{v} \Rrightarrow {\uparrow}Q'$, which holds by 1c.

(4) To show $\mathbf{fav}\,P = \mathbf{dom}\,(C)$ and C **singular with** $\widehat{\sigma}_0$ for some $\widehat{\sigma}_0$, we apply lemma 91 with $\Upsilon = \mathbf{fav}\,P = \mathbf{fav}(M')$ (as noted above, $\mathbf{dom}\,(C) \subseteq \mathbf{fav}(M') = \Upsilon$).
Now we will show that any substitution satisfying C is equivalent to $\widehat{\sigma}_1$. As noted in 1c, for any substitution $\Sigma \vdash \widehat{\sigma} : \Upsilon$, $\Sigma \vdash \widehat{\sigma} : C$ implies $T \vdash [\widehat{\sigma}]\,M' \simeq^{\leqslant} {\uparrow}Q$, which is rewritten as $T \vdash [\widehat{\sigma}]\,P \simeq^{\leqslant} Q$. And since $T \vdash [\widehat{\sigma}_1]\,P \simeq^{\leqslant} Q$, we have $T \vdash [\widehat{\sigma}]\,P \simeq^{\leqslant} [\widehat{\sigma}_1]\,P$, which implies $\Sigma \vdash \widehat{\sigma} \simeq^{\leqslant} \widehat{\sigma}_1 : \Upsilon$ by corollary 23.

(5) Let us show $T;\Gamma, x : [\widehat{\sigma}_0]\,P \vDash c' : \mathbf{nf}\,(N)$. By the soundness of singularity (lemma 90), we have $\Sigma \vdash \widehat{\sigma}_0 : C$, which by 1c means $T \vdash [\widehat{\sigma}_0]\,M' \simeq^{\leqslant} {\uparrow}Q$, that is $T \vdash [\widehat{\sigma}_0]\,P \simeq^{\leqslant} Q$, and thus, $T \vdash \Gamma, x : Q \simeq^{\leqslant} \Gamma, x : [\widehat{\sigma}_0]\,P$.
Then by lemma 48, $T;\Gamma, x : Q \vdash c' : N$ can be rewritten as $T;\Gamma, x : [\widehat{\sigma}_0]\,P \vdash c' : N$. Then by the induction hypothesis applied to it, $T;\Gamma, x : [\widehat{\sigma}_0]\,P \vDash c' : \mathbf{nf}\,(N)$ holds.

Case 13. $(\forall^{\text{INF}}_{\bullet\Rrightarrow})$

Since N cannot be a algorithmic variable, if $[\widehat{\sigma}]\,N$ starts with $\forall$, so does N. This way, $N = \forall\overrightarrow{\alpha^+}.\ N_1$. Then by assumption:

(1) $T \vdash^{\supseteq} \Sigma$

(2) $T\,;\mathbf{dom}\,(\Sigma) \vdash \forall\overrightarrow{\alpha^+}.\ N_1$ is free from negative algorithmic variables, and then $T, \overrightarrow{\alpha^+}\,;\mathbf{dom}\,(\Sigma) \vdash N_1$ is free from negative algorithmic variables too;

(3) $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\,N_1$;

(4) $T \vdash M$;

(5) $T\,;\Gamma \vdash [\widehat{\sigma}]\forall\overrightarrow{\alpha^+}.\ N_1 \bullet \overrightarrow{v} \Rrightarrow M$, that is $T\,;\Gamma \vdash (\forall\overrightarrow{\alpha^+}.\ [\widehat{\sigma}]\,N_1) \bullet \overrightarrow{v} \Rrightarrow M$. Then by inversion there exists σ such that
  (a) $T \vdash \sigma : \overrightarrow{\alpha^+}$;
  (b) $\overrightarrow{v} \neq \cdot$ and $\overrightarrow{\alpha^+} \neq \cdot$; and
  (c) $T\,;\Gamma \vdash [\sigma][\widehat{\sigma}]\,N_1 \bullet \overrightarrow{v} \Rrightarrow M$. Notice that σ and $\widehat{\sigma}$ commute because the codomain of σ does not contain algorithmic variables (and thus, does not intersect with the domain of $\widehat{\sigma}$), and the codomain of $\widehat{\sigma}$ is T and does not intersect with $\overrightarrow{\alpha^+}$—the domain of σ.
  Let us take fresh $\overrightarrow{\widehat{\alpha}^+}$ and construct $N_0 = [\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]\,N_1$ and $\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vdash \widehat{\sigma}_0 : \mathbf{fav}(N_0)$ defined as

$$\begin{cases} [\widehat{\sigma}_0]\widehat{\alpha}_i^+ = [\sigma]\alpha_i^+ & \text{for } \widehat{\alpha}_i^+ \in \overrightarrow{\widehat{\alpha}^+} \cap \mathbf{fav}\,N_0 \\ [\widehat{\sigma}_0]\widehat{\beta}^{\pm} = [\widehat{\sigma}]\widehat{\beta}^{\pm} & \text{for } \widehat{\beta}^{\pm} \in \mathbf{fav}\,N_1 \end{cases}$$

  Then it is easy to see that $[\widehat{\sigma}_0][\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]\,N_1 = [\sigma][\widehat{\sigma}]\,N_1$ because this substitution compositions coincide on $\mathbf{fav}(N_1) \cup \mathbf{fv}\,(N_1)$. In other words, $[\widehat{\sigma}_0]\,N_0 = [\sigma][\widehat{\sigma}]\,N_1$.
  Then let us apply the induction hypothesis to $T\,;\Gamma \vdash [\widehat{\sigma}_0]\,N_0 \bullet \overrightarrow{v} \Rrightarrow M$ and obtain M', Σ', and C such that
  - $T\,;\Gamma\,;\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vDash N_0 \bullet \overrightarrow{v} \Rrightarrow M' \dashv \Sigma'\,;C$ and
  - for any $\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vdash \widehat{\sigma}_0 : \mathbf{fav}(N_0)$ and $T \vdash M$ such that $T\,;\Gamma \vdash [\widehat{\sigma}_0]\,N_0 \bullet \overrightarrow{v} \Rightarrow$ $> M$, there exists $\widehat{\sigma}_0'$ such that
    (i) $\Sigma' \vdash \widehat{\sigma}_0' : \mathbf{fav}(N_0) \cup \mathbf{fav}(M')$, $\Sigma' \vdash \widehat{\sigma}_0' : C$,
    (ii) $\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vdash \widehat{\sigma}_0' \simeq^{\leqslant} \widehat{\sigma}_0 : \mathbf{fav}\,N_0$, and
    (iii) $T \vdash [\widehat{\sigma}_0']\,M' \simeq^{\leqslant} M$.

Let us take M', Σ', and C from the induction hypothesis (5c) (from C we subtract entries restricting $\overrightarrow{\widehat{\alpha}^+}$) and show they satisfy the required properties

(1) To infer $T\,;\Gamma\,;\Sigma \vDash \forall\overrightarrow{\alpha^+}.\ N_1 \bullet \vec{v} \Rrightarrow M' \dashv \Sigma'\,; C \setminus \overrightarrow{\widehat{\alpha}^+}$ we apply the corresponding algorithmic rule $(\forall^{\text{INF}}_{\bullet\Rrightarrow})$. As noted above, the required premises hold:

  (a) $\vec{v} \neq \cdot$, $\overrightarrow{\alpha^+} \neq \cdot$; and

  (b) $T\,;\Gamma\,;\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vDash [\overrightarrow{\widehat{\alpha}^+}/\overrightarrow{\alpha^+}]N_1 \bullet \vec{v} \Rrightarrow M' \dashv \Sigma'\,; C$ is obtained by unfolding the definition of N_0 in $T\,;\Gamma\,;\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vDash N_0 \bullet \vec{v} \Rrightarrow M' \dashv \Sigma'\,; C$ (5c).

(2) Let us take and arbitrary $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\,N_1$ and $T \vdash M$ and assume $T;\Gamma \vdash [\widehat{\sigma}]\forall\overrightarrow{\alpha^+}.\ N_1 \bullet \vec{v} \Rightarrow$ $>$ M. Then the same reasoning as in 5c applies. In particular, we construct $\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vdash \widehat{\sigma}_0 : \mathbf{fav}(N_0)$ as an extension of $\widehat{\sigma}$ and obtain $T\,;\Gamma \vdash [\widehat{\sigma}_0]N_0 \bullet \vec{v} \Rrightarrow M$.

It means we can apply the property inferred from the induction hypothesis (5c) to obtain $\widehat{\sigma}'_0$ such that

  (a) $\Sigma' \vdash \widehat{\sigma}'_0 : \mathbf{fav}(N_0) \cup \mathbf{fav}(M')$ and $\Sigma' \vdash \widehat{\sigma}'_0 : C$,

  (b) $\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vdash \widehat{\sigma}'_0 \simeq^{\leqslant} \widehat{\sigma}_0 : \mathbf{fav}\,N_0$, and

  (c) $T \vdash [\widehat{\sigma}'_0]M' \simeq^{\leqslant} M$.

Let us show that $\widehat{\sigma}'_0|_{(\mathbf{fav}(N_1)\cup\mathbf{fav}(M'))}$ satisfies the required properties.

  (a) $\Sigma' \vdash \widehat{\sigma}'_0|_{(\mathbf{fav}(N_1)\cup\mathbf{fav}(M'))} : (\mathbf{fav}(N_1) \cup \mathbf{fav}(M'))$ holds since $\Sigma' \vdash \widehat{\sigma}'_0 : \mathbf{fav}(N_0) \cup \mathbf{fav}(M')$ and $\mathbf{fav}(N_1)\cup\mathbf{fav}(M') \subseteq \mathbf{fav}(N_0)\cup\mathbf{fav}(M')$; $\Sigma' \vdash \widehat{\sigma}'_0|_{(\mathbf{fav}(N_1)\cup\mathbf{fav}(M'))} : C \setminus \overrightarrow{\widehat{\alpha}^+}$ holds since $\Sigma' \vdash \widehat{\sigma}'_0 : C$, $\Sigma' \vdash \widehat{\sigma}'_0 : \mathbf{fav}(N_0) \cup \mathbf{fav}(M')$, and $(\mathbf{fav}(N_0) \cup \mathbf{fav}(M')) \setminus \overrightarrow{\widehat{\alpha}^+} = \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$.

  (b) $T \vdash [\widehat{\sigma}'_0]M' \simeq^{\leqslant} M$ holds as shown, and hence it holds for $\widehat{\sigma}'_0|_{(\mathbf{fav}(N_1)\cup\mathbf{fav}(M'))}$;

  (c) We show $\Sigma \vdash \widehat{\sigma}'_0 \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}\,N_1$, from which it follows that it holds for $\widehat{\sigma}'_0|_{(\mathbf{fav}(N_1)\cup\mathbf{fav}(M'))}$. Let us take an arbitrary $\widehat{\beta}^{\pm} \in \mathbf{dom}\,(\Sigma) \subseteq \mathbf{dom}\,(\Sigma) \cup \overrightarrow{\widehat{\alpha}^+}$. Then since $\Sigma, \overrightarrow{\widehat{\alpha}^+}\{T\} \vdash \widehat{\sigma}'_0 \simeq^{\leqslant} \widehat{\sigma}_0 : \mathbf{fav}\,N_0$, we have $\Sigma(\widehat{\beta}^{\pm}) \vdash [\widehat{\sigma}'_0]\widehat{\beta}^{\pm} \simeq^{\leqslant} [\widehat{\sigma}_0]\widehat{\beta}^{\pm}$ and by definition of $\widehat{\sigma}_0$, $[\widehat{\sigma}_0]\widehat{\beta}^{\pm} = [\widehat{\sigma}]\widehat{\beta}^{\pm}$.

Case 14. $(\rightarrow^{\text{INF}}_{\bullet\Rrightarrow})$

Since N cannot be a algorithmic variable, if the shape of $[\widehat{\sigma}]N$ is an arrow, so is the shape of N. This way, $N = Q \rightarrow N_1$. Then by assumption:

(1) $T \vdash^{\supseteq} \Sigma$;

(2) $T\,; \mathbf{dom}\,(\Sigma) \vdash Q \rightarrow N_1$ is free from negative algorithmic variables;

(3) $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}\,Q \cup \mathbf{fav}\,N_1$;

(4) $T \vdash M$;

(5) $T\,;\Gamma \vdash [\widehat{\sigma}](Q \rightarrow N_1) \bullet v, \vec{v} \Rrightarrow M$, that is $T\,;\Gamma \vdash ([\widehat{\sigma}]Q \rightarrow [\widehat{\sigma}]N_1) \bullet v, \vec{v} \Rrightarrow M$, and by inversion:

  (a) $T;\Gamma \vdash v\colon P$, and by the induction hypothesis, $T;\Gamma \vDash v\colon \mathbf{nf}\,(P)$;

  (b) $T \vdash [\widehat{\sigma}]Q \geqslant P$, which by transitivity (lemma 22) means $T \vdash [\widehat{\sigma}]Q \geqslant \mathbf{nf}\,(P)$, and then by completeness of subtyping (lemma 79), $T;\,\Sigma \vDash Q \geqslant \mathbf{nf}\,(P) \dashv C_1$, for some $\Sigma \vdash C_1 : \mathbf{fav}(Q)$, and moreover, $\Sigma \vdash \widehat{\sigma} : C_1$;

  (c) $T\,;\Gamma \vdash [\widehat{\sigma}]N_1 \bullet \vec{v} \Rrightarrow M$. Notice that the induction hypothesis applies to this case: $T\,; \mathbf{dom}\,(\Sigma) \vdash N_1$ is free from negative algorithmic variables because so is $Q \rightarrow N_1$. This way, there exist M', Σ', and C_2 such that

    (i) $T\,;\Gamma\,;\Sigma \vDash N_1 \bullet \vec{v} \Rrightarrow M' \dashv \Sigma'\,; C_2$ and then by the soundness of typing (i.e. the induction hypothesis),

      (A) $\Sigma \subseteq \Sigma'$

      (B) $T\,; \mathbf{dom}\,(\Sigma') \vdash M'$

(C) $\mathbf{dom}\,(\Sigma) \cap \mathbf{fav}(M') \subseteq \mathbf{fav}\, N_1$
(D) $\Sigma'|_{\mathbf{fav}\,N_1 \cup \mathbf{fav}\,M'} \vdash C_2$

(ii) for any $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N_1)$ and $T \vdash M$ such that $T\,;\Gamma \vdash [\widehat{\sigma}]N_1 \bullet \vec{v} \Rrightarrow M$, there exists $\widehat{\sigma}'$ such that
(A) $\Sigma' \vdash \widehat{\sigma}' : \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$ and $\Sigma' \vdash \widehat{\sigma}' : C_2$,
(B) $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}(N_1)$, and
(C) $T \vdash [\widehat{\sigma}']M' \simeq^{\leqslant} M$.

We need to show that there exist M', Σ', and C such that $T\,;\Gamma\,;\Sigma \vDash Q \to N_1 \bullet v, \vec{v} \Rightarrow > M' \dashv \Sigma'\,; C$ and the initiality property holds. We take M' and Σ' from the induction hypothesis (5c), and C as a merge of C_1 and C_2. To show that $\Sigma' \vdash C_1 \,\&\, C_2 = C$ exists, we apply lemma 83. To do so, we need to provide a substitution satisfying both C_1 and C_2.
Notice that $\mathbf{dom}\,(C_1) = \mathbf{fav}(Q)$ and $\mathbf{dom}\,(C_2) \subseteq \mathbf{fav}\,N_1 \cup \mathbf{fav}\,M'$. This way, it suffices to construct $\Sigma' \vdash \widehat{\sigma}'' : \mathbf{fav}(Q) \cup \mathbf{fav}\,N_1 \cup \mathbf{fav}\,M'$ such that $\Sigma' \vdash \widehat{\sigma}'' : C_1$ and $\Sigma' \vdash \widehat{\sigma}'' : C_2$.
By the induction hypothesis (5(c)ii), $\widehat{\sigma}|_{\mathbf{fav}(N_1)}$ can be extended to $\widehat{\sigma}'$ such that
(1) $\Sigma' \vdash \widehat{\sigma}' : \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$ and $\Sigma' \vdash \widehat{\sigma}' : C_2$,
(2) $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}(N_1)$, and
(3) $T \vdash [\widehat{\sigma}']M' \simeq^{\leqslant} M$.

Let us extend $\widehat{\sigma}'$ to $\widehat{\sigma}''$ defined on $\mathbf{fav}(Q) \cup \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$ with values of $\widehat{\sigma}$ as follows:

$$\begin{cases} [\widehat{\sigma}'']\widehat{\beta}^{\pm} = [\widehat{\sigma}']\widehat{\beta}^{\pm} & \text{for } \widehat{\beta}^{\pm} \in \mathbf{fav}(N_1) \cup \mathbf{fav}(M') \\ [\widehat{\sigma}'']\widehat{\gamma}^{\pm} = [\widehat{\sigma}]\widehat{\gamma}^{\pm} & \text{for } \widehat{\gamma}^{\pm} \in \mathbf{fav}(Q) \setminus (\mathbf{fav}(N_1) \cup \mathbf{fav}(M')) \end{cases}$$

First, notice that $\Sigma' \vdash \widehat{\sigma}'' \simeq^{\leqslant} \widehat{\sigma}' : \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$ by definition. Then since $\Sigma' \vdash \widehat{\sigma}' : C_2$ and $\Sigma' \vdash C_2 : \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$, we have $\Sigma' \vdash \widehat{\sigma}'' : C_2$.
Second, notice that $\Sigma \vdash \widehat{\sigma}'' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}(N_1) \cup \mathbf{fav}(Q)$:

- if $\widehat{\gamma}^{\pm} \in \mathbf{fav}(Q) \setminus (\mathbf{fav}(N_1) \cup \mathbf{fav}(M'))$ then $[\widehat{\sigma}'']\widehat{\gamma}^{\pm} = [\widehat{\sigma}]\widehat{\gamma}^{\pm}$ by definition of $\widehat{\sigma}''$;
- if $\widehat{\gamma}^{\pm} \in \mathbf{fav}(Q) \cap \mathbf{fav}(N_1)$ then $[\widehat{\sigma}'']\widehat{\gamma}^{\pm} = [\widehat{\sigma}']\widehat{\gamma}^{\pm}$, and $\Sigma \vdash \widehat{\sigma}' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}(N_1)$, as noted above;
- if $\widehat{\gamma}^{\pm} \in \mathbf{fav}(Q) \cap \mathbf{fav}(M')$ then since $T\,;\mathbf{dom}\,(\Sigma) \vdash Q$, we have $\mathbf{fav}(Q) \subseteq \mathbf{dom}\,(\Sigma)$, implying $\widehat{\gamma}^{\pm} \in \mathbf{dom}\,(\Sigma) \cap \mathbf{fav}(M') \subseteq \mathbf{fav}(N_1)$. This way, $\widehat{\gamma}^{\pm} \in \mathbf{fav}(Q) \cap \mathbf{fav}(N_1)$, and this case is covered by the previous one.

In particular, $\Sigma \vdash \widehat{\sigma}'' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}(Q)$. Then since $\Sigma \vdash \widehat{\sigma} : C_1$ and $\Sigma \vdash C_1 : \mathbf{fav}(Q)$, we have $\Sigma \vdash \widehat{\sigma}'' : C_1$.
This way, $\widehat{\sigma}'$ satisfies both C_1 and C_2, and by the completeness of constraint merge (lemma 83), $\Sigma' \vdash C_1 \,\&\, C_2 = C$ exists.
Finally, to show the required properties, we take M' and Σ' from the induction hypothesis (5(c)ii), and C defined above. Then

(1) $T\,;\Gamma\,;\Sigma \vDash Q \to N_1 \bullet v, \vec{v} \Rrightarrow M' \dashv \Sigma'\,; C$ is inferred by $(\to^{\text{INF}}_{\bullet\Rrightarrow})$. As noted above:
    (a) $T; \Gamma \vDash v\colon \mathbf{nf}\,(P)$,
    (b) $T; \Sigma \vDash Q \geqslant \mathbf{nf}\,(P) \dashv C_1$,
    (c) $T\,;\Gamma\,;\Sigma \vDash N_1 \bullet \vec{v} \Rrightarrow M' \dashv \Sigma'\,; C_2$, and
    (d) $\Sigma' \vdash C_1 \,\&\, C_2 = C$.
(2) let us take an arbitrary $\Sigma \vdash \widehat{\sigma}_0 : \mathbf{fav}\,Q \cup \mathbf{fav}\,N_1$; and $T \vdash M_0$; such that $T\,;\Gamma \vdash [\widehat{\sigma}_0](Q \to N_1) \bullet v, \vec{v} \Rrightarrow M_0$. Then by inversion of $T\,;\Gamma \vdash [\widehat{\sigma}_0]Q \to [\widehat{\sigma}_0]N_1 \bullet v, \vec{v} \Rrightarrow M_0$, we have the same properties as in 5. In particular,
    - $T \vdash [\widehat{\sigma}_0]Q \geqslant \mathbf{nf}\,(P)$ and by the completeness of subtyping (lemma 79), $\Sigma \vdash \widehat{\sigma}_0 : C_1$.
    - $T\,;\Gamma \vdash [\widehat{\sigma}_0]N_1 \bullet \vec{v} \Rrightarrow M_0$. Then by 5(c)ii, there exists $\widehat{\sigma}_0'$ such that

(a) $\Sigma' \vdash \widehat{\sigma}'_0 : \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$ and $\Sigma' \vdash \widehat{\sigma}'_0 : C_2$,
(b) $\Sigma \vdash \widehat{\sigma}'_0 \simeq^{\leqslant} \widehat{\sigma}_0 : \mathbf{fav}(N_1)$, and
(c) $T \vdash [\widehat{\sigma}'_0]M' \simeq^{\leqslant} M_0$.

Let us extend $\widehat{\sigma}'_0$ to be defined on $\mathbf{fav}(Q) \cup \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$ with the values of $\widehat{\sigma}_0$. We define $\widehat{\sigma}''_0$ as follows:

$$\begin{cases} [\widehat{\sigma}''_0]\widehat{\gamma}^{\pm} = [\widehat{\sigma}'_0]\widehat{\gamma}^{\pm} & \text{for } \widehat{\gamma}^{\pm} \in \mathbf{fav}(N_1) \cup \mathbf{fav}(M') \\ [\widehat{\sigma}''_0]\widehat{\gamma}^{\pm} = [\widehat{\sigma}_0]\widehat{\gamma}^{\pm} & \text{for } \widehat{\gamma}^{\pm} \in \mathbf{fav}(Q) \setminus (\mathbf{fav}(N_1) \cup \mathbf{fav}(M')) \end{cases}$$

This way,
- $\Sigma' \vdash \widehat{\sigma}''_0 : \mathbf{fav}(Q) \cup \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$,
- $\Sigma' \vdash \widehat{\sigma}''_0 : C$, since $\Sigma' \vdash \widehat{\sigma}''_0 : C_1$ and $\Sigma' \vdash \widehat{\sigma}''_0 : C_2$, which is proved similarly to $\Sigma' \vdash \widehat{\sigma}'' : C_1$ and $\Sigma' \vdash \widehat{\sigma}'' : C_2$ above;
- $\Sigma \vdash \widehat{\sigma}''_0 \simeq^{\leqslant} \widehat{\sigma}_0 : \mathbf{fav}(N_1) \cup \mathbf{fav}(Q)$: the proof is analogous to $\Sigma \vdash \widehat{\sigma}'' \simeq^{\leqslant} \widehat{\sigma} : \mathbf{fav}(N_1) \cup \mathbf{fav}(Q)$ above.
- $T \vdash [\widehat{\sigma}''_0]M' \simeq^{\leqslant} M_0$ Notice that $\Sigma' \vdash \widehat{\sigma}''_0 \simeq^{\leqslant} \widehat{\sigma}'_0 : \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$, which is proved analogously to $\Sigma' \vdash \widehat{\sigma}'' \simeq^{\leqslant} \widehat{\sigma}' : \mathbf{fav}(N_1) \cup \mathbf{fav}(M')$ above. Then $T \vdash [\widehat{\sigma}'_0]M' \simeq^{\leqslant} M_0$ can be rewritten to $T \vdash [\widehat{\sigma}''_0]M' \simeq^{\leqslant} M_0$.

Case 15. ($\emptyset^{\text{INF}}_{\bullet\Longrightarrow}$)
By assumption:
(1) $T \vdash^{\supseteq} \Sigma$,
(2) $T \vdash N'$,
(3) $T ; \mathbf{dom}\,(\Sigma) \vdash N$ and N is free from negative variables,
(4) $\Sigma \vdash \widehat{\sigma} : \mathbf{fav}(N)$,
(5) $T ; \Gamma \vdash [\widehat{\sigma}]N \bullet \cdot \Longrightarrow N'$, and by inversion, $T \vdash [\widehat{\sigma}]N \simeq^{\leqslant} N'$.

Then we can apply the corresponding algorithmic rule ($\emptyset^{\text{INF}}_{\bullet\Longrightarrow}$) to infer $T ; \Gamma ; \Sigma \vDash N \bullet \cdot \Longrightarrow \mathbf{nf}\,(N) \dashv \Sigma ; \cdot$. Let us show the required properties. Let us take an arbitrary $\Sigma \vdash \widehat{\sigma}_0 : \mathbf{fav}(N)$ and $T \vdash M$ such that $T ; \Gamma \vdash [\widehat{\sigma}_1]N \bullet \cdot \Longrightarrow M$. Then we can take $\widehat{\sigma}_0$ as the required substitution:
(1) $\Sigma \vdash \widehat{\sigma}_0 : \mathbf{fav}(N) \cup \mathbf{fav}(\mathbf{nf}\,(N))$, since $\mathbf{fav}(\mathbf{nf}\,(N)) = \mathbf{fav}(N)$, and thus, $\mathbf{fav}(N) \cup \mathbf{fav}(\mathbf{nf}\,(N)) = \mathbf{fav}(N)$;
(2) $\Sigma \vdash \widehat{\sigma}_0 : \cdot$ vacuously;
(3) $\Sigma \vdash \widehat{\sigma}_0 \simeq^{\leqslant} \widehat{\sigma}_0 : \mathbf{fav}(N)$ by reflexivity;
(4) Let us show $T \vdash [\widehat{\sigma}_0]\mathbf{nf}\,(N) \simeq^{\leqslant} M$. Notice that $T ; \Gamma \vdash [\widehat{\sigma}_0]N \bullet \cdot \Longrightarrow M$ can only be inferred by ($\emptyset^{\text{INF}}_{\bullet\Longrightarrow}$), and thus, $T \vdash [\widehat{\sigma}_0]N \simeq^{\leqslant} M$. By corollary 17, $T \vdash [\widehat{\sigma}_0]N \simeq^{\leqslant} [\widehat{\sigma}_0]\mathbf{nf}\,(N)$, and then by transitivity, $T \vdash [\widehat{\sigma}_0]\mathbf{nf}\,(N) \simeq^{\leqslant} M$, that is $T \vdash [\widehat{\sigma}_0]\mathbf{nf}\,(N) \simeq^{\leqslant} M$.

□