Formalizing CoreGI

Technical Report No. 248

Stefan Wehr
University of Freiburg, Germany
wehr@informatik.uni-freiburg.de

July 21, 2009

Preface

This document contains the full formalization of CoreGI, a subset of JavaGI.[1] All typing rules are fully worked out and all proofs are complete. However, there is no explaining text yet. For questions, please contact the author.

Contents

[1] http://www.informatik.uni-freiburg.de/~wehr/javagi/

$$
\begin{aligned}
\mathit{prog} &::= \overline{\mathit{def}}\ e \\
\mathit{def} &::= \mathit{cdef} \mid \mathit{idef} \mid \mathit{impl} \\
\mathit{cdef} &::= \texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P} \\
&\qquad \{\ \overline{T\ f}\ \overline{m : \mathit{mdef}}\ \} \\
\mathit{idef} &::= \texttt{interface}\ I\langle\overline{X}\rangle\ [\overline{Y\ \texttt{where}\ \overline{R}}]\ \texttt{where}\ \overline{P} \\
&\qquad \{\ \overline{m : \texttt{static}\ \mathit{msig}}\ \overline{\mathit{rcsig}}\ \} \\
\mathit{impl} &::= \texttt{implementation}\langle\overline{X}\rangle\ K\ [\overline{N}]\ \texttt{where}\ \overline{P} \\
&\qquad \{\ \overline{\texttt{static}\ \mathit{mdef}}\ \overline{\mathit{rcdef}}\ \} \\
\mathit{rcsig} &::= \texttt{receiver}\,\{\overline{m : \mathit{msig}}\} \\
\mathit{rcdef} &::= \texttt{receiver}\,\{\overline{\mathit{mdef}}\} \\
\mathit{msig} &::= \langle\overline{X}\rangle\,\overline{T\ x} \to T\ \texttt{where}\ \overline{P} \\
\mathit{mdef} &::= \mathit{msig}\,\{e\} \\
M, N &::= C\langle\overline{T}\rangle \mid \texttt{Object} \\
G, H &::= X \mid N \\
K, L &::= I\langle\overline{T}\rangle \\
T, U, V, W &::= G \mid K \\
R, S &::= \overline{G}\ \texttt{implements}\ K \\
\mathcal{R}, \mathcal{S} &::= \overline{T}\ \texttt{implements}\ K \\
P, Q &::= R \mid X\ \texttt{extends}\ T \\
\mathcal{P}, \mathcal{Q} &::= \mathcal{R} \mid T\ \texttt{extends}\ T \\
e &::= x \mid e.f \mid e.m\langle\overline{T}\rangle(\overline{e}) \mid K[\overline{T}].m\langle\overline{T}\rangle(\overline{e}) \\
&\quad \mid \texttt{new}\ N(\overline{e}) \mid (N)\,e \\
X, Y, Z &\in \mathit{TvarName} \quad C, D \in \mathit{ClsName} \quad I, J \in \mathit{IfaceName} \\
m &\in \mathit{MethodName} \quad f, g \in \mathit{FieldName} \quad x, y \in \mathit{VarName}
\end{aligned}
$$

Figure 1: Syntax of CoreGI.

1 Definition of CoreGI

$$\boxed{N \trianglelefteq_{\mathtt{c}} N}$$

EXT-C-REFL
$$N \trianglelefteq_{\mathtt{c}} N$$

EXT-C-SUPER
$$\frac{\texttt{class } C\langle\overline{X}\rangle \texttt{ extends } M \ldots \qquad [\overline{T/X}]M \trianglelefteq_{\mathtt{c}} N}{C\langle\overline{T}\rangle \trianglelefteq_{\mathtt{c}} N}$$

$$\boxed{K \trianglelefteq_{\mathtt{i}} K}$$

EXT-I-REFL
$$K \trianglelefteq_{\mathtt{i}} K$$

EXT-I-SUPER
$$\frac{\texttt{interface } I\langle\overline{X}\rangle\, [Y \texttt{ where } \overline{R}] \ldots \qquad R_i = Y \texttt{ implements } K \qquad [\overline{T/X}]K \trianglelefteq_{\mathtt{i}} L}{I\langle\overline{T}\rangle \trianglelefteq_{\mathtt{i}} L}$$

Figure 2: Class and interface inheritance.

$$\boxed{\mathsf{getmdef}^{\mathsf{c}}(m, N) = \langle\overline{X}\rangle\,\overline{T\,x} \to T\ \mathtt{where}\ \overline{\mathcal{P}}\,\{e\}}$$

DYN-MDEF-C-BASE

$$\frac{\mathtt{class}\ C\langle\overline{X}\rangle\ \mathtt{extends}\ N\ \mathtt{where}\ \overline{P}\,\{\,\overline{T\,f}\ \ \overline{m : \mathit{mdef}}\,\}}{\mathsf{getmdef}^{\mathsf{c}}(m_j, C\langle\overline{U}\rangle) = [\overline{U/X}]\mathit{mdef}_j}$$

DYN-MDEF-C-SUPER

$$\frac{\begin{array}{c}\mathtt{class}\ C\langle\overline{X}\rangle\ \mathtt{extends}\ N\ \mathtt{where}\ \overline{P}\,\{\,\overline{T\,f}\ \ \overline{m : \mathit{mdef}}\,\}\\ m \notin \overline{m} \qquad \mathsf{getmdef}^{\mathsf{c}}(m, [\overline{U/X}]N) = \langle\overline{X}\rangle\,\overline{V\,x} \to V\ \mathtt{where}\ \overline{\mathcal{P}}\,\{e\}\end{array}}{\mathsf{getmdef}^{\mathsf{c}}(m, C\langle\overline{U}\rangle) = \langle\overline{X}\rangle\,\overline{V\,x} \to V\ \mathtt{where}\ \overline{\mathcal{P}}\,\{e\}}$$

$$\boxed{\mathsf{getmdef}^{\mathsf{i}}(m, N, \overline{N}) = \langle\overline{X}\rangle\,\overline{T\,x} \to T\ \mathtt{where}\ \overline{\mathcal{P}}\,\{e\}}$$

DYN-MDEF-I

$$\frac{\begin{array}{c}\mathtt{interface}\ I\langle\overline{Z'}\rangle\,[\overline{Z}^{l}\ \mathtt{where}\,\overline{R}]\ \mathtt{where}\ \overline{P}\,\{\ldots\ \overline{\mathit{rcsig}}\,\}\\ \mathit{rcsig}_j = \mathtt{receiver}\,\{\overline{m : \mathit{msig}}\} \qquad \mathit{msig}_k = \langle\overline{Y}\rangle\,\overline{T\,x} \to T\ \mathtt{where}\ \overline{Q}\\ (\forall i \in [l], i \neq j)\ \mathsf{contrib}_{Z_i}(\overline{T}, \overline{N}) = M_i^{?} \qquad \mathsf{contrib}_{Z_j}(Z_j\overline{T}, N\overline{N}) = M_j^{?}\\ \mathsf{minimpl}\{([\overline{V/X}], \mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{M'}\,]\ \ldots) \mid (\forall i \in [l])\ M_i^{?} = \mathsf{nil}\ \text{or}\ M_i^{?} \trianglelefteq_{\mathsf{c}} [\overline{V/X}]M_i'\}\\ = (\sigma, \mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{M'}\,]\ \mathtt{where}\ \overline{P'}\ \{\ldots\ \overline{\mathit{rcdef}}\,\})\\ \mathit{rcdef}_j = \mathtt{receiver}\,\{\overline{\mathit{mdef}}\}\end{array}}{\mathsf{getmdef}^{\mathsf{i}}(m_k, N, \overline{N}^{n}) = \sigma\,\mathit{mdef}_k}$$

$$\boxed{\mathsf{getsmdef}(m, K, \overline{U}) = \langle\overline{X}\rangle\,\overline{T\,x} \to T\ \mathtt{where}\ \overline{\mathcal{P}}\,\{e\}}$$

DYN-MDEF-S

$$\frac{\begin{array}{c}\mathtt{interface}\ I\langle\overline{Z'}\rangle\,[\overline{Z}\,\mathtt{where}\,\overline{R}]\ \mathtt{where}\ \overline{Q}\,\{\,\overline{m : \mathtt{static}\,\mathit{msig}}\ \ldots\}\\ (\sigma, \mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{N}^{l}\,]\ \mathtt{where}\ \overline{P}\ \{\,\overline{\mathtt{static}\,\mathit{mdef}}\ \ldots\}) =\\ \mathsf{minimpl}\{([\overline{V/X}], \mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{N}^{l}\,]\ \ldots) \mid (\forall i \in [l])\ N_i = \mathtt{Object}\ \text{or}\ W_i \trianglelefteq_{\mathsf{c}} [\overline{V/X}]N_i\}\end{array}}{\mathsf{getsmdef}(m_k, I\langle\overline{T}\rangle, \overline{W}^{l}) = \sigma\,\mathit{mdef}_k}$$

$$\boxed{\mathsf{minimpl}\{\overline{(\sigma, \mathit{impl})}\} = (\sigma, \mathit{impl}) \quad \mathsf{contrib}_X(\overline{T}, \overline{N}) = N^{?} \quad N \sqcup N = N \quad \bigsqcup \mathscr{N} = N}$$

MIN-MDEF

$$\frac{\mathit{impl}_i = \mathtt{implementation}\langle\overline{X_i}\rangle\ I\langle\overline{V_i}\rangle\ [\,\overline{N_i}^{l}\,]\ \ldots \qquad n \geq 1 \qquad (\forall i \in [n])\ \sigma_k\overline{N_k} \trianglelefteq_{\mathsf{c}} \sigma_i\overline{N_i}}{\mathsf{minimpl}\{(\sigma_1, \mathit{impl}_1), \ldots, (\sigma_n, \mathit{impl}_n)\} = (\sigma_k, \mathit{impl}_k)}$$

CONTRIB-NON-EMPTY

$$\frac{\mathscr{C} = \{N_i \mid i \in [n], T_i = X\} \qquad \mathscr{C} \neq \emptyset \qquad \bigsqcup \mathscr{C} = M}{\mathsf{contrib}_X(\overline{T}^{n}, \overline{N}^{n}) = M}$$

CONTRIB-EMPTY

$$\frac{\mathscr{C} = \{N_i \mid i \in [n], T_i = X\} \qquad \mathscr{C} = \emptyset}{\mathsf{contrib}_X(\overline{T}^{n}, \overline{N}^{n}) = \mathsf{nil}}$$

LUB-RIGHT

$$\frac{N \trianglelefteq_{\mathsf{c}} M}{N \sqcup M = M}$$

LUB-LEFT

$$\frac{M \trianglelefteq_{\mathsf{c}} N}{N \sqcup M = N}$$

LUB-SUPER

$$\frac{\begin{array}{c}\text{not}\ C\langle\overline{T}\rangle \trianglelefteq_{\mathsf{c}} N \qquad \text{not}\ N \trianglelefteq_{\mathsf{c}} C\langle\overline{T}\rangle\\ \mathtt{class}\ C\langle\overline{X}\rangle\ \mathtt{extends}\ N'\ \ldots \qquad [\overline{T/X}]N' \sqcup N = M\end{array}}{C\langle\overline{T}\rangle \sqcup N = M}$$

LUB-SET-SINGLE

$$\bigsqcup \{N\} = N$$

LUB-SET-MULTI

$$\frac{\mathscr{N} \neq \emptyset \qquad \bigsqcup \mathscr{N} = M' \qquad M' \sqcup N = M}{\bigsqcup (\mathscr{N} \mathbin{\dot\cup} \{N\}) = M}$$

Figure 3: Dynamic selection of method definitions.

Values and evaluation contexts

$$
\begin{array}{rcl}
v, w & ::= & \texttt{new}\, N(\overline{v}) \\
\mathcal{E} & ::= & \Box \mid \mathcal{E}.f \mid \mathcal{E}.m\langle\overline{T}\rangle(\overline{e}) \mid v.m\langle\overline{T}\rangle(\overline{v}, \mathcal{E}, \overline{e}) \\
& \mid & K[\overline{T}].m\langle\overline{T}\rangle(\overline{v}, \mathcal{E}, \overline{e}) \mid \texttt{new}\, N(\overline{v}, \mathcal{E}, \overline{e}) \mid (N)\, \mathcal{E}
\end{array}
$$

Top-level reduction: $e \longmapsto e$

$$
\textsc{dyn-field}\ \frac{\mathsf{fields}(N) = \overline{U\, f}}{\texttt{new}\, N(\overline{v}).f_i \longmapsto v_i}
\qquad
\textsc{dyn-invoke-c}\ \frac{v = \texttt{new}\, N(\overline{w}) \qquad \mathsf{getmdef}^{\mathrm{c}}(m^{\mathrm{c}}, N) = \langle\overline{X}\rangle\, \overline{T\, x} \to T\ \texttt{where}\ \overline{\mathcal{P}}\, \{e\}}{v.m^{\mathrm{c}}\langle\overline{U}\rangle(\overline{v}) \longmapsto [v/\texttt{this}, \overline{v/x}][\overline{U/X}]e}
$$

$$
\textsc{dyn-invoke-i}\ \frac{(\forall i \in \{0, \ldots, n\})\ v_i = \texttt{new}\, N_i(\overline{w_i}) \qquad \mathsf{getmdef}^{\mathrm{i}}(m^{\mathrm{i}}, N_0, \overline{N}) = \langle\overline{X}\rangle\, \overline{T\, x} \to T\ \texttt{where}\ \overline{\mathcal{P}}\, \{e\}}{v_0.m^{\mathrm{i}}\langle\overline{U}\rangle(\overline{v}^n) \longmapsto [v_0/\texttt{this}, \overline{v/x}][\overline{U/X}]e}
$$

$$
\textsc{dyn-invoke-s}\ \frac{\mathsf{getsmdef}(m, K, \overline{U}) = \langle\overline{X}\rangle\, \overline{T\, x} \to T\ \texttt{where}\ \overline{\mathcal{P}}\, \{e\}}{K[\overline{U}].m\langle\overline{V}\rangle(\overline{v}) \longmapsto [\overline{v/x}][\overline{V/X}]e}
\qquad
\textsc{dyn-cast}\ \frac{v = \texttt{new}\, M(\overline{w}) \qquad M \trianglelefteq_{\mathrm{c}} N}{(N)\, v \longmapsto v}
$$

Context reduction: $e \longrightarrow e$

$$
\textsc{dyn-context}\ \frac{e \longmapsto e'}{\mathcal{E}[e] \longrightarrow \mathcal{E}[e']}
$$

$\mathsf{fields}(N) = \overline{T\, f}$

$$
\textsc{fields-object}\ \mathsf{fields}(\texttt{Object}) = \bullet
\qquad
\textsc{fields-class}\ \frac{\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\, \{\overline{T\, f} \ldots\} \qquad \mathsf{fields}([\overline{U/X}]N) = \overline{T'\, f'}}{\mathsf{fields}(C\langle\overline{U}\rangle) = \overline{T'\, f'}, [\overline{U/X}]\overline{T\, f}}
$$

Figure 4: Dynamic semantics of CoreGI.

$\mathsf{non\text{-}static}(I)$

$$
\textsc{non-static-iface}\ \frac{\begin{array}{c}\texttt{interface}\ I\langle\overline{X}\rangle\, [\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\, \{\overline{m : \texttt{static}\ msig}^n\ \ldots\} \\ n = 0 \qquad (\forall i)\ \text{if } R_i = \overline{Z}\ \texttt{implements}\ J\langle\overline{T}\rangle \text{ then } \mathsf{non\text{-}static}(J)\end{array}}{\mathsf{non\text{-}static}(I)}
$$

$j \in \mathsf{pos}^{\pi}(I) \quad X \in \mathsf{pos}^{\pi}(msig) \quad X \in \mathsf{pos}^{\pi}(rcsig) \quad X \in \mathsf{pos}^{\pi}(Q)$

$$
\textsc{pos-iface}\ \frac{\begin{array}{c}\texttt{interface}\ I\langle\overline{X}\rangle\, [\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\, \{\overline{m : \texttt{static}\ msig}\ \overline{rcsig}\} \\ (\forall i)\ Y_j \in \mathsf{pos}^{\pi}(msig_i) \qquad (\forall i)\ Y_j \in \mathsf{pos}^{\pi}(rcsig_i) \qquad (\forall i)\ Y_j \in \mathsf{pos}^{\pi}(R_i) \qquad Y_j \notin \mathsf{ftv}(\overline{P})\end{array}}{j \in \mathsf{pos}^{\pi}(I)}
$$

$$
\textsc{pos-msig-plus}\ \frac{Y \notin \mathsf{ftv}(\overline{T})}{Y \in \mathsf{pos}^{+}(\langle\overline{X}\rangle\, \overline{T\, x} \to U\ \texttt{where}\ \overline{P})}
\qquad
\textsc{pos-msig-minus}\ \frac{Y \notin \mathsf{ftv}(U)}{Y \in \mathsf{pos}^{-}(\langle\overline{X}\rangle\, \overline{T\, x} \to U\ \texttt{where}\ \overline{P})}
$$

$$
\textsc{pos-recv}\ \frac{(\forall i)\ X \in \mathsf{pos}^{\pi}(msig_i)}{X \in \mathsf{pos}^{\pi}(\texttt{receiver}\, \{\overline{m : msig}\})}
\qquad
\textsc{pos-constr}\ \frac{(\forall i)\ \text{if } X = G_i \text{ then } i \in \mathsf{pos}^{\pi}(I)}{X \in \mathsf{pos}^{\pi}(\overline{G}\ \texttt{implements}\ I\langle\overline{U}\rangle)}
$$

Figure 5: Restrictions on interfaces and implementing types.

$$\boxed{\Delta \Vdash \mathcal{P}}$$

$$\textsc{ent-extends}\ \frac{\Delta \vdash T \le U}{\Delta \Vdash T\ \texttt{extends}\ U} \qquad \textsc{ent-env}\ \frac{P \in \Delta}{\Delta \Vdash P} \qquad \textsc{ent-super}\ \frac{\texttt{interface}\ I\langle\overline{X}\rangle\ [\overline{Y}\ \texttt{where}\ \overline{R}]\ \ldots \qquad \Delta \Vdash \overline{U}\ \texttt{implements}\ I\langle\overline{T}\rangle}{\Delta \Vdash [\overline{T/X}, \overline{U/Y}]R_i}$$

$$\textsc{ent-impl}\ \frac{\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots \qquad \Delta \Vdash [\overline{U/X}]\overline{P}}{\Delta \Vdash [\overline{U/X}](\overline{N}\ \texttt{implements}\ I\langle\overline{T}\rangle)}$$

$$\textsc{ent-up}\ \frac{\Delta \vdash U \le U' \qquad \Delta \Vdash \overline{T}\, U'\, \overline{V}\ \texttt{implements}\ I\langle\overline{W}\rangle \qquad n \in \mathsf{pos}^-(I)}{\Delta \Vdash \overline{T}^{n-1}\, U\, \overline{V}\ \texttt{implements}\ I\langle\overline{W}\rangle} \qquad \textsc{ent-iface}\ \frac{1 \in \mathsf{pos}^+(I) \qquad \mathsf{non\text{-}static}(I)}{\Delta \Vdash I\langle\overline{T}\rangle\ \texttt{implements}\ I\langle\overline{T}\rangle}$$

$$\boxed{\Delta \vdash T \le T}$$

$$\textsc{sub-refl}\ \ \Delta \vdash T \le T \qquad \textsc{sub-object}\ \ \Delta \vdash T \le \texttt{Object} \qquad \textsc{sub-trans}\ \frac{\Delta \vdash S \le T \qquad \Delta \vdash T \le U}{\Delta \vdash S \le U} \qquad \textsc{sub-var}\ \frac{X\ \texttt{extends}\ T \in \Delta}{\Delta \vdash X \le T}$$

$$\textsc{sub-class}\ \frac{\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N \ldots}{\Delta \vdash C\langle\overline{T}\rangle \le [\overline{T/X}]N} \qquad \textsc{sub-iface}\ \frac{\texttt{interface}\ I\langle\overline{X}\rangle\ [Y\ \texttt{where}\ \overline{R}]\ \ldots \qquad R_i = Y\ \texttt{implements}\ K}{\Delta \vdash I\langle\overline{T}\rangle \le [\overline{T/X}]K}$$

$$\textsc{sub-impl}\ \frac{\Delta \Vdash T\ \texttt{implements}\ K}{\Delta \vdash T \le K}$$

Figure 6: Entailment and subtyping.

$$\boxed{\mathsf{mtype}_\Delta(m, T) = \langle\overline{X}\rangle\, \overline{U\ x} \to U\ \texttt{where}\ \mathcal{P} \qquad \mathsf{smtype}_\Delta(m, K[\overline{T}]) = \langle\overline{X}\rangle\, \overline{U\ x} \to U\ \texttt{where}\ \mathcal{P}}$$

$$\textsc{mtype-class}\ \frac{\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\, \{\ldots\ \overline{m : msig\,\{e\}}\,\}}{\mathsf{mtype}_\Delta(m^{\mathrm{c}}_j, C\langle\overline{T}\rangle) = [\overline{T/X}]\, msig_j}$$

$$\textsc{mtype-iface}\ \frac{\begin{array}{c}\texttt{interface}\ I\langle\overline{X}\rangle\ [\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\, \{\ldots\ \overline{rcsig}\,\} \\ rcsig_j = \texttt{receiver}\, \{\overline{m : msig}\} \qquad \Delta \Vdash \overline{T}\ \texttt{implements}\ I\langle\overline{V}\rangle\end{array}}{\mathsf{mtype}_\Delta(m^{\mathrm{i}}_k, T_j) = [\overline{V/X}, \overline{T/Y}]\, msig_k}$$

$$\textsc{mtype-static}\ \frac{\texttt{interface}\ I\langle\overline{X}\rangle\ [\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\, \{\,\overline{m : \texttt{static}\ msig}\ \ldots\} \qquad \Delta \Vdash \overline{T}\ \texttt{implements}\ I\langle\overline{U}\rangle}{\mathsf{smtype}_\Delta(m^{\mathrm{i}}_k, I\langle\overline{U}\rangle[\overline{T}]) = [\overline{U/X}, \overline{T/Y}]\, msig_k}$$

Figure 7: Method types.

$\boxed{\Delta \vdash T\ \mathsf{ok}}$

$$\frac{X \in \mathsf{dom}(\Delta)}{\Delta \vdash X\ \mathsf{ok}}\ \textsc{ok-tvar} \qquad \Delta \vdash \texttt{Object}\ \mathsf{ok}\ \textsc{ok-object}$$

$$\frac{\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\ \ldots \qquad \Delta \vdash \overline{T}\ \mathsf{ok} \qquad \Delta \Vdash [\overline{T/X}]\overline{P}}{\Delta \vdash C\langle\overline{T}\rangle\ \mathsf{ok}}\ \textsc{ok-class}$$

$$\frac{\begin{array}{c}\texttt{interface}\ I\langle\overline{X}\rangle\,[Y\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\ \ldots\\ \Delta \vdash \overline{T}\ \mathsf{ok} \qquad Y \notin \mathsf{ftv}(\overline{T},\Delta) \qquad \Delta, Y\ \texttt{implements}\ I\langle\overline{T}\rangle \Vdash [\overline{T/X}]\overline{R},\overline{P}\end{array}}{\Delta \vdash I\langle\overline{T}\rangle\ \mathsf{ok}}\ \textsc{ok-iface}$$

$\boxed{\Delta \vdash \mathcal{P}\ \mathsf{ok}}$

$$\frac{\texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\ \ldots \qquad \Delta \vdash \overline{T},\overline{U}\ \mathsf{ok} \qquad \Delta \Vdash [\overline{U/X},\overline{T/Y}]\overline{R},\overline{P}}{\Delta \vdash \overline{T}\ \texttt{implements}\ I\langle\overline{U}\rangle\ \mathsf{ok}}\ \textsc{ok-impl-constr}$$

$$\frac{\Delta \vdash T, U\ \mathsf{ok}}{\Delta \vdash T\ \texttt{extends}\ U\ \mathsf{ok}}\ \textsc{ok-ext-constr}$$

Figure 8: Well-formedness of types and constraints.

$\boxed{\Delta;\Gamma \vdash e : T}$

$$\Delta;\Gamma \vdash x : \Gamma(x)\ \textsc{exp-var} \qquad \frac{\Delta;\Gamma \vdash e : C\langle\overline{T}\rangle \qquad \texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\,\{\,\overline{U\,f}\ldots\}}{\Delta;\Gamma \vdash e.f_j : [\overline{T/X}]U_j}\ \textsc{exp-field}$$

$$\frac{\begin{array}{c}\Delta;\Gamma \vdash e : T \qquad \mathsf{mtype}_\Delta(m,T) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}\\ (\forall i)\ \Delta;\Gamma \vdash e_i : [\overline{V/X}]U_i \qquad \Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}} \qquad \Delta \vdash \overline{V}\ \mathsf{ok}\end{array}}{\Delta;\Gamma \vdash e.m\langle\overline{V}\rangle(\overline{e}) : [\overline{V/X}]U}\ \textsc{exp-invoke}$$

$$\frac{\begin{array}{c}\mathsf{smtype}_\Delta(m, I\langle\overline{W}\rangle[\overline{T}]) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}\\ (\forall i)\ \Delta;\Gamma \vdash e_i : [\overline{V/X}]U_i \qquad \Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}} \qquad \Delta \vdash \overline{T},\overline{V}\ \mathsf{ok}\end{array}}{\Delta;\Gamma \vdash I\langle\overline{W}\rangle[\overline{T}].m\langle\overline{V}\rangle(\overline{e}) : [\overline{V/X}]U}\ \textsc{exp-invoke-s}$$

$$\frac{\Delta \vdash N\ \mathsf{ok} \qquad \mathsf{fields}(N) = \overline{T\,f} \qquad (\forall i)\ \Delta;\Gamma \vdash e_i : T_i}{\Delta;\Gamma \vdash \texttt{new}\,N(\overline{e}) : N}\ \textsc{exp-new} \qquad \frac{\Delta \vdash N\ \mathsf{ok} \qquad \Delta;\Gamma \vdash e : T}{\Delta;\Gamma \vdash (N)\,e : N}\ \textsc{exp-cast}$$

$$\frac{\Delta;\Gamma \vdash e : U \qquad \Delta \vdash U \leq T}{\Delta;\Gamma \vdash e : T}\ \textsc{exp-subsume}$$

Figure 9: Expression typing.

$\boxed{\Delta \vdash msig \leq msig \quad \mathsf{override\text{-}ok}_\Delta(m : msig, N)}$

SUB-MSIG
$$\frac{\Delta, \overline{P} \vdash T \leq T'}{\Delta \vdash \langle\overline{X}\rangle\,\overline{T\,x} \to T \;\texttt{where}\; \overline{P} \leq \langle\overline{X}\rangle\,\overline{T\,x} \to T' \;\texttt{where}\; \overline{P}}$$

OK-OVERRIDE
$$\frac{\begin{array}{c}(\forall N')\ \text{if}\ \Delta \vdash N \leq N'\ \text{and}\ \mathsf{mtype}_\Delta(m, N') = \langle\overline{X}\rangle\,\overline{T\,x} \to T \;\texttt{where}\; \overline{\mathcal{P}}\\ \text{then}\ \Delta \vdash msig \leq \langle\overline{X}\rangle\,\overline{T\,x} \to T \;\texttt{where}\; \overline{\mathcal{P}}\end{array}}{\mathsf{override\text{-}ok}_\Delta(m : msig, N)}$$

$\boxed{\Delta \vdash msig\ \mathsf{ok} \quad \Delta; \Gamma \vdash mdef\ \mathsf{ok} \quad \Delta \vdash m : mdef\ \mathsf{ok\,in}\, N \quad \Delta \vdash rcsig\ \mathsf{ok}}$

OK-MSIG
$$\frac{\Delta, \overline{P}, \overline{X} \vdash \overline{T}, U, \overline{P}\ \mathsf{ok}}{\Delta \vdash \langle\overline{X}\rangle\,\overline{T\,x} \to U \;\texttt{where}\; \overline{P}\ \mathsf{ok}}$$

OK-MDEF
$$\frac{\Delta \vdash \langle\overline{X}\rangle\,\overline{T\,x} \to U \;\texttt{where}\; \overline{P}\ \mathsf{ok} \qquad \Delta, \overline{P}, \overline{X}; \Gamma, \overline{x : T} \vdash e : U}{\Delta; \Gamma \vdash \langle\overline{X}\rangle\,\overline{T\,x} \to U \;\texttt{where}\; \overline{P}\,\{e\}\ \mathsf{ok}}$$

OK-MDEF-IN-CLASS
$$\frac{\Delta; \texttt{this} : N \vdash msig\,\{e\}\ \mathsf{ok} \qquad \mathsf{override\text{-}ok}_\Delta(m : msig, N)}{\Delta \vdash m : msig\,\{e\}\ \mathsf{ok\,in}\, N}$$

OK-RCSIG
$$\frac{(\forall i)\ \Delta \vdash msig_i\ \mathsf{ok}}{\Delta \vdash \texttt{receiver}\,\{\overline{m : msig}\}\ \mathsf{ok}}$$

$\boxed{\Delta \vdash mdef\ \mathsf{implements}\ msig \quad \Delta \vdash rcdef\ \mathsf{implements}\ rcsig}$

IMPL-METH
$$\frac{\Delta; \Gamma \vdash msig\,\{e\}\ \mathsf{ok} \qquad \Delta \vdash msig \leq msig'}{\Delta; \Gamma \vdash msig\,\{e\}\ \mathsf{implements}\ msig'}$$

IMPL-RECV
$$\frac{(\forall i)\ \Delta; \Gamma \vdash mdef_i\ \mathsf{implements}\ msig_i}{\Delta; \Gamma \vdash \texttt{receiver}\,\{\overline{mdef}\}\ \mathsf{implements}\ \texttt{receiver}\,\{\overline{m : msig}\}}$$

$\boxed{\vdash cdef\ \mathsf{ok} \quad \vdash idef\ \mathsf{ok} \quad \vdash impl\ \mathsf{ok}}$

OK-CDEF
$$\frac{\overline{P}, \overline{X} \vdash N, \overline{P}, \overline{T}\ \mathsf{ok} \qquad (\forall i)\ \overline{P}, \overline{X} \vdash m_i : mdef_i\ \mathsf{ok\,in}\, C\langle\overline{X}\rangle}{\vdash \texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\,\{\,\overline{T\,f}\ \overline{m : mdef}\,\}\ \mathsf{ok}}$$

OK-IDEF
$$\frac{\overline{R}, \overline{P}, \overline{X}, \overline{Y} \vdash \overline{R}, \overline{P}, \overline{msig}, \overline{rcsig}\ \mathsf{ok}}{\vdash \texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\,\{\,\overline{m : \texttt{static}\ msig}\ \overline{rcsig}\,\}\ \mathsf{ok}}$$

OK-IMPL
$$\frac{\begin{array}{c}\overline{P}, \overline{X} \vdash \overline{N}\ \texttt{implements}\ I\langle\overline{T}\rangle, \overline{P}\ \mathsf{ok}\\ \texttt{interface}\ I\langle\overline{Y}\rangle\,[\overline{Z}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{Q}\,\{\,\overline{m : \texttt{static}\ msig}\ \overline{rcsig}\,\}\\ (\forall i)\ \overline{P}, \overline{X}; \emptyset \vdash mdef_i\ \mathsf{implements}\ [\overline{T/Y}, \overline{N/Z}]msig_i\\ (\forall i)\ \overline{P}, \overline{X}; \texttt{this} : N_i \vdash rcdef_i\ \mathsf{implements}\ [\overline{T/Y}, \overline{N/Z}]rcsig_i\end{array}}{\vdash \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \{\,\overline{\texttt{static}\ mdef}\ \overline{rcdef}\,\}\ \mathsf{ok}}$$

$\boxed{\vdash prog\ \mathsf{ok}}$

OK-PROG
$$\frac{\vdash \overline{def}\ \mathsf{ok} \qquad \emptyset; \emptyset \vdash e : T}{\vdash \overline{def}\ e\ \mathsf{ok}}$$

Figure 10: Program typing.

2 Quasi-algorithmic Subtyping and Entailment

$\boxed{\Delta \Vdash_{\mathrm{q}} \mathcal{P}}$

$$\frac{\Delta \vdash_{\mathrm{q}} T \leq U}{\Delta \Vdash_{\mathrm{q}} T \; \texttt{extends} \; U} \; \textsc{ent-q-alg-extends}$$

$$\frac{(\forall i) \; \Delta \vdash_{\mathrm{q}}{}' \; T_i \leq U_i \qquad (\forall i) \text{ if } T_i \neq U_i \text{ then } i \in \mathsf{pos}^-(I) \qquad \Delta \Vdash_{\mathrm{q}}{}' \; \overline{U} \; \texttt{implements} \; I\langle\overline{V}\rangle}{\Delta \Vdash_{\mathrm{q}} \overline{T} \; \texttt{implements} \; I\langle\overline{V}\rangle} \; \textsc{ent-q-alg-up}$$

$\boxed{\Delta \Vdash_{\mathrm{q}}{}' \; \mathcal{R}}$

$$\frac{S \in \Delta \qquad R \in \mathsf{sup}(S)}{\Delta \Vdash_{\mathrm{q}}{}' \; R} \; \textsc{ent-q-alg-env}$$

$$\frac{\texttt{implementation}\langle\overline{X}\rangle \; I\langle\overline{T}\rangle \; [\,\overline{N}\,] \; \texttt{where} \; \overline{P} \ldots \qquad \Delta \Vdash_{\mathrm{q}} [\overline{U/X}]\overline{P}}{\Delta \Vdash_{\mathrm{q}}{}' \; [\overline{U/X}](\overline{N} \; \texttt{implements} \; I\langle\overline{T}\rangle)} \; \textsc{ent-q-alg-impl}$$

$$\frac{1 \in \mathsf{pos}^+(I) \qquad I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} K \qquad \mathsf{non\text{-}static}(I)}{\Delta \Vdash_{\mathrm{q}}{}' \; I\langle\overline{V}\rangle \; \texttt{implements} \; K} \; \textsc{ent-q-alg-iface}$$

$\boxed{\mathcal{R} \in \mathsf{sup}(\mathcal{R})}$

$$\mathcal{R} \in \mathsf{sup}(\mathcal{R}) \; \textsc{sup-id}$$

$$\frac{\texttt{interface} \; I\langle\overline{X}\rangle \; [\overline{Y} \; \texttt{where} \; \overline{S}] \ldots \qquad \overline{U} \; \texttt{implements} \; I\langle\overline{V}\rangle \in \mathsf{sup}(\mathcal{R})}{[\overline{V/X}, \overline{U/Y}]S_j \in \mathsf{sup}(\mathcal{R})} \; \textsc{sup-step}$$

Figure 11: Quasi-algorithmic entailment.

$$\boxed{\Delta \vdash_{\mathrm{q}}{}' \ T \leq T}$$

$$\frac{}{\Delta \vdash_{\mathrm{q}}{}' \ T \leq \texttt{Object}}\ \textsc{sub-q-alg-obj} \qquad \frac{}{\Delta \vdash_{\mathrm{q}}{}' \ X \leq X}\ \textsc{sub-q-alg-var-refl}$$

$$\frac{X \ \texttt{extends} \ T \in \Delta \qquad U \neq X, U \neq \texttt{Object} \qquad \Delta \vdash_{\mathrm{q}}{}' \ T \leq U}{\Delta \vdash_{\mathrm{q}}{}' \ X \leq U}\ \textsc{sub-q-alg-var} \qquad \frac{N \trianglelefteq_{\mathrm{c}} N' \qquad N' \neq \texttt{Object}}{\Delta \vdash_{\mathrm{q}}{}' \ N \leq N'}\ \textsc{sub-q-alg-class}$$

$$\frac{K \trianglelefteq_{\mathrm{i}} K'}{\Delta \vdash_{\mathrm{q}}{}' \ K \leq K'}\ \textsc{sub-q-alg-iface}$$

$$\boxed{\Delta \vdash_{\mathrm{q}}{}' \ T \leq T}$$

$$\frac{\Delta \vdash_{\mathrm{q}}{}' \ T \leq U}{\Delta \vdash_{\mathrm{q}} T \leq U}\ \textsc{sub-q-alg-kernel} \qquad \frac{\Delta \vdash_{\mathrm{q}}{}' \ T \leq U \qquad \Delta \Vdash_{\mathrm{q}}{}' \ U \ \texttt{implements} \ K}{\Delta \vdash_{\mathrm{q}} T \leq K}\ \textsc{sub-q-alg-impl}$$

Figure 12: Quasi-algorithmic subtyping.

3 Entailment and Subtyping Algorithms

$$\boxed{\Delta \Vdash_{\mathrm{a}} \mathcal{P}}$$

$$\textsc{ent-alg-main}\quad \frac{\Delta; \emptyset; \texttt{false} \Vdash_{\mathrm{a}} \mathcal{P}}{\Delta \Vdash_{\mathrm{a}} \mathcal{P}}$$

$$\boxed{\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \mathcal{P}}$$

$$\textsc{ent-alg-extends}\quad \frac{\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq U}{\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} T\,\texttt{extends}\,U}$$

$$\textsc{ent-alg-env}\quad \frac{R \in \Delta \qquad \overline{G}\,\texttt{implements}\,I\langle\overline{V}\rangle \in \mathsf{sup}(R) \qquad \Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{G}}{\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \overline{T}\,\texttt{implements}\,I\langle\overline{V}\rangle}$$

$$\textsc{ent-alg-iface}_1\quad \frac{\Delta; \beta; I \vdash_{\mathrm{a}} T \uparrow I\langle\overline{V}\rangle \qquad 1 \in \mathsf{pos}^{+}(I) \qquad \mathsf{non\text{-}static}(I)}{\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} T\,\texttt{implements}\,I\langle\overline{V}\rangle}$$

$$\textsc{ent-alg-iface}_2\quad \frac{1 \in \mathsf{pos}^{+}(I) \qquad I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} K \qquad \mathsf{non\text{-}static}(I)}{\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} I\langle\overline{V}\rangle\,\texttt{implements}\,K}$$

$$\textsc{ent-alg-impl}\quad \frac{\begin{array}{c}\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots \qquad \Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow [\overline{U/X}]\overline{N} \qquad \overline{V} = [\overline{U/X}]\overline{V'} \\ {[\overline{U/X}]\overline{N}}\,\texttt{implements}\,I\langle\overline{V}\rangle \notin \mathscr{G} \qquad \Delta; \mathscr{G} \cup \{[\overline{U/X}]\overline{N}\,\texttt{implements}\,I\langle\overline{V}\rangle\}; \texttt{false} \Vdash_{\mathrm{a}} [\overline{U/X}]\overline{P}\end{array}}{\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \overline{T}\,\texttt{implements}\,I\langle\overline{V}\rangle}$$

$$\boxed{\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{U}}$$

$$\textsc{ent-alg-lift}\quad \frac{(\forall i)\ \Delta \vdash_{\mathrm{a}}{}' T_i \leq U_i \qquad \beta \text{ or } \big((\forall i) \text{ if } T_i \neq U_i \text{ then } i \in \mathsf{pos}^{-}(I)\big)}{\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T}^{n} \uparrow \overline{U}^{n}}$$

$$\boxed{\Delta \vdash_{\mathrm{a}}{}' T \leq U \quad \Delta \vdash_{\mathrm{a}} T \leq U}$$

$$\textsc{sub-alg-kernel-quasi}\quad \frac{\Delta \vdash_{\mathrm{q}}{}' T \leq U}{\Delta \vdash_{\mathrm{a}}{}' T \leq U} \qquad\qquad \textsc{sub-alg-main}\quad \frac{\Delta; \emptyset \vdash_{\mathrm{a}} T \leq U}{\Delta \vdash_{\mathrm{a}} T \leq U}$$

$$\boxed{\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq U}$$

$$\textsc{sub-alg-kernel}\quad \frac{\Delta \vdash_{\mathrm{a}}{}' T \leq U}{\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq U} \qquad\qquad \textsc{sub-alg-impl}\quad \frac{\Delta; \mathscr{G}; \texttt{true} \Vdash_{\mathrm{a}} T\,\texttt{implements}\,K}{\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq K}$$

Figure 13: Algorithmic entailment and subtyping.

$$\boxed{\{\overline{T_i \le^? U_i}\} \xRightarrow{\Delta} \{\overline{T_i' \le^? U_i}\}}$$

$$\text{U-CLASS}\quad \frac{C \neq D \qquad \texttt{class}\ C\langle\overline{Y}\rangle\ \texttt{extends}\ M\ \ldots}{\{C\langle\overline{T}\rangle \le^? D\langle\overline{U}\rangle\} \,\dot{\cup}\, \mathscr{S} \xRightarrow{\Delta} \{[\overline{T/Y}]M \le^? D\langle\overline{U}\rangle\} \cup \mathscr{S}}$$

$$\text{U-IFACE-UP}\quad \frac{I \neq J \qquad \texttt{interface}\ I\langle\overline{X}\rangle\ [Y\ \texttt{where}\ \overline{R}]\ \ldots \qquad R_i = Y\ \texttt{implements}\ K}{\{I\langle\overline{T}\rangle \le^? J\langle\overline{U}\rangle\} \,\dot{\cup}\, \mathscr{S} \xRightarrow{\Delta} \{[\overline{T/X}]K \le^? J\langle\overline{U}\rangle\} \cup \mathscr{S}}$$

$$\text{U-IFACE-OBJECT}\quad \frac{}{\{K \le^? G\} \,\dot{\cup}\, \mathscr{S} \xRightarrow{\Delta} \{\texttt{Object} \le^? G\} \cup \mathscr{S}}$$

$$\text{U-VAR-ENV}\quad \frac{X\ \texttt{extends}\ T \in \Delta}{\{X \le^? U\} \,\dot{\cup}\, \mathscr{S} \xRightarrow{\Delta} \{T \le^? G\} \cup \mathscr{S}}$$

$$\text{U-VAR-OBJECT}\quad \frac{X\ \texttt{extends}\ T \notin \Delta \text{ for all } T}{\{X \le^? U\} \,\dot{\cup}\, \mathscr{S} \xRightarrow{\Delta} \{\texttt{Object} \le^? U\} \cup \mathscr{S}}$$

Figure 14: Transformation rules for unification modulo subtyping

```
unify_≤(Δ, X̄, {T_i ≤? U_i}‾) {
  for (each normal form {V_i ≤? W_i}‾ of {T_i ≤? U_i}‾ according to =Δ=>) {
    if (unify_=({V_i =? W_i}‾, X̄) == OK(σ))
      return OK(σ);
  }
  return FAIL;
}
```

```
unify_=({T_i =? U_i}‾, X̄) {
  if (there exists an idempotent mgu σ of {T_i =? U_i}‾ with dom(σ) ⊆ X̄)
    return OK(σ);
  else
    return FAIL;
}
```

Figure 15: Algorithm for unification modulo subtyping

```
entails(Δ,𝒫) { return entailsAux(Δ,∅,false,𝒫); }
entailsAux(Δ,𝒢,β,𝒫) {
  switch (𝒫) {
    case T extends U:
      return subAux(Δ,𝒢,T,U);
    case X mono: return X mono ∈ Δ;
    case N mono: return true;
    case T̄ implements I⟨V̄⟩:
      // rule ENT-ALG-ENV
      for (R ∈ Δ, Ḡ implements I⟨V̄⟩ ∈ sup(R)) { if (lift(Δ,β,I,T̄,Ḡ)) return true; }
      switch (T̄) {
        // rule ENT-ALG-IFACE1
        case T: if (lift(Δ,β,I,T,I⟨V̄⟩) && 1 ∈ pos⁺(I) && non-static(I)) return true;
        // rule ENT-ALG-IFACE2
        case J⟨W̄⟩: if (1 ∈ pos⁺(J) && J⟨W̄⟩ ⊴ᵢ I⟨V̄⟩ && non-static(J)) return true;
      }
      // rule ENT-ALG-IMPL
      for implementation⟨X̄⟩ I⟨W̄⟩ [N̄] where P̄ⁿ ... {
        if (unify≤(Δ,X̄,{T_i ≤? N_i}) == OK(σ) && lift(Δ,β,I,T̄,σN̄)
            && V̄==σW̄ && σN̄ implements I⟨V̄⟩ ∉ 𝒢) {
          𝒢₀ = 𝒢 ∪ {σN̄ implements I⟨V̄⟩};
          if (∀i ∈ [n],entailsAux(Δ,𝒢₀,false,σP_i)) return true;
        }
      }
      return false;      // no rule applicable
  }
}

sub(Δ,T,U) { return subAux(Δ,∅,T,U); }
subAux(Δ,𝒢,T,U) {
  if (sub'(Δ,T,U)) return true;
  switch (U) { case K: return entailsAux(Δ,𝒢,true,T implements K); }
  return false;
}

sub'(Δ,T,U) {
  switch (T,U) {
    case (_,Object): return true;
    case (X,X): return true;
    case (X,_):
      for X extends V ∈ Δ { if (sub'(Δ,V,U)) return true; }
      return false;
    case (N₁,N₂): return N₁ ⊴c N₂;
    case (K₁,K₂): return K₁ ⊴ᵢ K₂;
  }
  return false;
}

lift(Δ,β,I,T̄ⁿ,Ūᵐ) {
  return (n==m && ∀i ∈ [n],(sub'(Δ,T_i,U_i) && (β || T_i==U_i || i ∈ pos⁻(I))));
}
```

Figure 16: Entailment and subtyping algorithms.

4 Expression Typing Algorithm

$\boxed{\Delta \Vdash^?_{\mathrm{a}} \overline{T^?}\ \texttt{implements}\ I\langle\overline{T^?}\rangle \twoheadrightarrow \mathcal{R}}$

$$\text{ENT-NIL-ALG-MAIN}\quad \frac{\Delta;\emptyset;\texttt{false} \Vdash^?_{\mathrm{a}} \overline{T^?}\ \texttt{implements}\ I\langle\overline{U^?}\rangle \twoheadrightarrow \mathcal{R}}{\Delta \Vdash^?_{\mathrm{a}} \overline{T^?}\ \texttt{implements}\ I\langle\overline{U^?}\rangle \twoheadrightarrow \mathcal{R}}$$

$\boxed{\Delta;\mathscr{G};\beta \Vdash^?_{\mathrm{a}} \overline{T^?}\ \texttt{implements}\ I\langle\overline{T^?}\rangle \twoheadrightarrow \mathcal{R}}$

$$\text{ENT-NIL-ALG-ENV}\quad \frac{R \in \Delta \qquad \overline{G}\ \texttt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(R) \qquad \Delta;\beta;I \vdash^?_{\mathrm{a}} \overline{T^?} \uparrow \overline{G} \twoheadrightarrow \overline{T} \qquad (\forall i)\ V^?_i \sharp V_i}{\Delta;\mathscr{G};\beta \Vdash^?_{\mathrm{a}} \overline{T^?}\ \texttt{implements}\ I\langle\overline{V^?}\rangle \twoheadrightarrow \overline{T}\ \texttt{implements}\ I\langle\overline{V}\rangle}$$

$$\text{ENT-NIL-ALG-IFACE}_1\quad \frac{\Delta;\beta;I \vdash_{\mathrm{a}} T \uparrow I\langle\overline{V}\rangle \qquad 1 \in \mathsf{pos}^+(I) \qquad \mathsf{non\text{-}static}(I) \qquad (\forall i)\ V^?_i \sharp V_i}{\Delta;\mathscr{G};\beta \Vdash^?_{\mathrm{a}} T\ \texttt{implements}\ I\langle\overline{V^?}\rangle \twoheadrightarrow T\ \texttt{implements}\ I\langle\overline{V}\rangle}$$

$$\text{ENT-NIL-ALG-IFACE}_2\quad \frac{1 \in \mathsf{pos}^+(I) \qquad \mathsf{non\text{-}static}(I) \qquad I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} J\langle\overline{U}\rangle \qquad (\forall i)\ U^?_i \sharp U_i}{\Delta;\mathscr{G};\beta \Vdash^?_{\mathrm{a}} I\langle\overline{V}\rangle\ \texttt{implements}\ J\langle\overline{U^?}\rangle \twoheadrightarrow I\langle\overline{V}\rangle\ \texttt{implements}\ J\langle\overline{U}\rangle}$$

$$\text{ENT-NIL-ALG-IMPL}\quad \frac{\begin{array}{c}\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots \\ \Delta;\beta;I \vdash^?_{\mathrm{a}} \overline{T^?} \uparrow \overline{[U/X]}\overline{N} \twoheadrightarrow \overline{T} \qquad (\forall i)\ V^?_i \sharp \overline{[U/X]}V_i \qquad \overline{[U/X]}\overline{N}\ \texttt{implements}\ I\langle\overline{[U/X]}\overline{V}\rangle \notin \mathscr{G} \\ \Delta;\mathscr{G} \cup \{\overline{[U/X]}\overline{N}\ \texttt{implements}\ I\langle\overline{[U/X]}\overline{V}\rangle\};\texttt{false} \Vdash_{\mathrm{a}} \overline{[U/X]}\overline{P}\end{array}}{\Delta;\mathscr{G};\beta \Vdash^?_{\mathrm{a}} \overline{T^?}\ \texttt{implements}\ I\langle\overline{V^?}\rangle \twoheadrightarrow \overline{T}\ \texttt{implements}\ I\langle\overline{[U/X]}\overline{V}\rangle}$$

$\boxed{\Delta;\beta;I \vdash^?_{\mathrm{a}} \overline{T^?} \uparrow \overline{T} \twoheadrightarrow \overline{T}}$

$$\text{ENT-NIL-ALG-LIFT}\quad \frac{\begin{array}{c}(\forall i)\ T^?_i = \mathsf{nil} \text{ or } \Delta \vdash_{\mathrm{a}}{}' T^?_i \leq U_i \qquad \beta \text{ or } \big((\forall i) \text{ if } T^?_i \neq U_i \text{ and } T^?_i \neq \mathsf{nil} \text{ then } i \in \mathsf{pos}^-(I)\big) \\ (\forall i)\ \ \text{if } T^?_i = \mathsf{nil} \text{ then } V_i = U_i \text{ else } V_i = T^?_i\end{array}}{\Delta;\beta;I \vdash^?_{\mathrm{a}} \overline{T^?}^n \uparrow \overline{U}^n \twoheadrightarrow \overline{V}^n}$$

$\boxed{T^? \sharp T}$

$$\text{MATCHES-NIL}\quad \mathsf{nil} \sharp T \qquad\qquad \text{MATCHES-EQUAL}\quad T \sharp T$$

Figure 17: Algorithmic entailment for nillable constraints.

$$\boxed{\textsf{a-mtype}_\Delta(m, T, \overline{T}) = \langle \overline{X} \rangle\, \overline{U\,x} \to U \ \texttt{where}\ \overline{\mathcal{P}}}$$

ALG-MTYPE-CLASS

$$\frac{\textsf{bound}_\Delta(T) = N \qquad \textsf{a-mtype}^{\mathrm{c}}(m^{\mathrm{c}}, N) = \langle \overline{X} \rangle\, \overline{U\,x} \to U \ \texttt{where}\ \overline{\mathcal{P}}}{\textsf{a-mtype}_\Delta(m^{\mathrm{c}}, T, \overline{T}) = \langle \overline{X} \rangle\, \overline{U\,x} \to U \ \texttt{where}\ \overline{\mathcal{P}}}$$

ALG-MTYPE-IFACE

$$\frac{\begin{array}{c}\texttt{interface}\ I\langle \overline{Z'} \rangle\, [\overline{Z}^{l}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\, \{ \ldots\ \overline{rcsig}\, \} \\ rcsig_j = \texttt{receiver}\, \{\overline{m : msig}\} \qquad m^{\mathrm{i}} = m_k \qquad msig_k = \langle \overline{Y} \rangle\, \overline{U\,x} \to U\ \texttt{where}\ \overline{Q} \\ (\forall i \in [l], i \neq j)\ \textsf{contrib}'_{\Delta;Z_i}(\overline{U}, \overline{T}) = \mathscr{V}_i^{?} \qquad \textsf{contrib}'_{\Delta;Z_j}(Z_j\,\overline{U}, T\,\overline{T}) = \mathscr{V}_j^{?} \\ p^{?} = (\text{if } U = Z_i \text{ for some } i \in [l] \text{ then } i \text{ else } \textsf{nil}) \\ \overline{W}\ \texttt{implements}\ I\langle \overline{W'} \rangle = \\ \textsf{pick-constr}_\Delta^{p^{?}}\{\overline{V}\ \texttt{implements}\ I\langle \overline{V''} \rangle \mid (\forall i \in [l])\ \text{if } \mathscr{V}_i^{?} = \textsf{nil} \text{ then } V_i^{?} = \textsf{nil} \\ \text{else define } V_i^{?} \text{ such that} \\ \Delta \vdash_{\mathrm{a}}{}' V_i' \leq V_i^{?} \text{ for some } V_i' \in \mathscr{V}_i^{?}, \\ \Delta \Vdash_{\mathrm{a}}^{?} \overline{V^{?}}\ \texttt{implements}\ I\langle \overline{\textsf{nil}} \rangle \twoheadrightarrow \overline{V}\ \texttt{implements}\ I\langle \overline{V''} \rangle\} \end{array}}{\textsf{a-mtype}_\Delta(m^{\mathrm{i}}, T, \overline{T}) = [\overline{W/Z}, \overline{W'/Z'}]\, msig_k}$$

$$\boxed{\textsf{a-mtype}^{\mathrm{c}}(m, N) = \langle \overline{X} \rangle\, \overline{U\,x} \to U \ \texttt{where}\ \overline{\mathcal{P}}}$$

ALG-MTYPE-DIRECT

$$\frac{\texttt{class}\ C\langle \overline{X} \rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\, \{\, \overline{T\,f}\ \overline{m : mdef}\, \} \qquad mdef_i = msig\,\{e\}}{\textsf{a-mtype}^{\mathrm{c}}(m_i, C\langle \overline{T} \rangle) = [\overline{T/X}]\, msig}$$

ALG-MTYPE-SUPER

$$\frac{\begin{array}{c}\texttt{class}\ C\langle \overline{X} \rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\, \{\, \overline{T\,f}\ \overline{m : mdef}\, \} \\ m \notin \overline{m} \qquad \textsf{a-mtype}^{\mathrm{c}}(m, [\overline{T/X}]N) = \langle \overline{X} \rangle\, \overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}\end{array}}{\textsf{a-mtype}^{\mathrm{c}}(m, C\langle \overline{T} \rangle) = \langle \overline{X} \rangle\, \overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}}$$

$$\boxed{\textsf{bound}_\Delta(T) = N \quad \textsf{pick-constr}_\Delta^{k^{?}} \mathscr{R} = \mathcal{R} \quad \textsf{contrib}'_{\Delta;X}(\overline{T}, \overline{T}) = \mathscr{T}^{?} \quad \textsf{MUB}_\Delta(\mathscr{T}) = \mathscr{T}}$$

BOUND-VAR

$$\frac{\Delta \vdash_{\mathrm{a}}{}' X \leq N \qquad \text{if } \Delta \vdash_{\mathrm{a}}{}' X \leq N' \text{ then } N \trianglelefteq_{\mathrm{c}} N'}{\textsf{bound}_\Delta(X) = N}$$

BOUND-CLASS

$$\textsf{bound}_\Delta(N) = N$$

BOUND-IFACE

$$\textsf{bound}_\Delta(K) = \texttt{Object}$$

PICK-CONSTR-NIL

$$\frac{n \geq 1 \qquad i \in [n]}{\textsf{pick-constr}_\Delta^{\textsf{nil}}\{\overline{\mathcal{R}}^{n}\} = \mathcal{R}_i}$$

PICK-CONSTR-NON-NIL

$$\frac{n \geq 1 \qquad (\forall i \in [n])\ \Delta \vdash_{\mathrm{a}}{}' T_{jk} \leq T_{ik}}{\textsf{pick-constr}_\Delta^{k}\{\overline{T_1}\ \texttt{implements}\ K_1, \ldots, \overline{T_n}\ \texttt{implements}\ K_n\} = \overline{T_j}\ \texttt{implements}\ K_j}$$

CONTRIB-NON-EMPTY'

$$\frac{\mathscr{T} = \{T_i \mid i \in [n], U_i = X\} \qquad \mathscr{T} \neq \emptyset \qquad \mathscr{V} = \textsf{MUB}_\Delta(\mathscr{T})}{\textsf{contrib}'_{\Delta;X}(\overline{U}^{n}, \overline{T}^{n}) = \mathscr{V}}$$

CONTRIB-EMPTY'

$$\frac{\{T_i \mid i \in [n], U_i = X\} = \emptyset}{\textsf{contrib}'_{\Delta;X}(\overline{U}^{n}, \overline{T}^{n}) = \textsf{nil}}$$

MUB

$$\frac{\mathscr{V} = \{V \mid (\forall T \in \mathscr{T}), \Delta \vdash_{\mathrm{a}}{}' T \leq V\} \qquad \mathscr{U} = \{V \in \mathscr{V} \mid (\forall V' \in \mathscr{V} \setminus \{V\}) \text{ not } \Delta \vdash_{\mathrm{a}}{}' V' \leq V\}}{\textsf{MUB}_\Delta(\mathscr{T}) = \mathscr{U}}$$

Figure 18: Algorithmic method types.

$\boxed{\Delta; \Gamma \vdash_{\mathrm{a}} e : T}$

$$\textsc{exp-alg-var}\quad \Delta; \Gamma \vdash_{\mathrm{a}} x : \Gamma(x)$$

$$\textsc{exp-alg-field}\quad \frac{\Delta; \Gamma \vdash_{\mathrm{a}} e : T \qquad \mathsf{bound}_{\Delta}(T) = N \qquad \mathsf{fields}(N) = \overline{U\, f}}{\Delta; \Gamma \vdash_{\mathrm{a}} e.f_j : U_j}$$

$$\textsc{exp-alg-invoke-d}\quad \frac{\begin{array}{c}\Delta; \Gamma \vdash_{\mathrm{a}} e : T \qquad (\forall i)\ \Delta; \Gamma \vdash_{\mathrm{a}} e_i : T_i \qquad \mathsf{a\text{-}mtype}_{\Delta}(m, T, \overline{T}) = \langle \overline{X} \rangle\, \overline{U\, x} \rightarrow U \ \mathtt{where}\ \overline{\mathcal{P}} \\ (\forall i)\ \Delta \vdash_{\mathrm{a}} T_i \leq [\overline{V/X}]U_i \qquad \Delta \Vdash_{\mathrm{a}} [\overline{V/X}]\overline{\mathcal{P}} \qquad \Delta \vdash_{\mathrm{a}} \overline{V}\ \mathsf{ok}\end{array}}{\Delta; \Gamma \vdash_{\mathrm{a}} e.m\langle \overline{V} \rangle(\overline{e}) : [\overline{V/X}]U}$$

$$\textsc{exp-alg-invoke-s}\quad \frac{\begin{array}{c}\mathsf{a\text{-}smtype}_{\Delta}(m, I\langle \overline{W} \rangle[\overline{T}]) = \langle \overline{X} \rangle\, \overline{U\, x} \rightarrow U \ \mathtt{where}\ \overline{\mathcal{P}} \\ (\forall i)\ \Delta; \Gamma \vdash_{\mathrm{a}} e_i : U_i' \qquad (\forall i)\ \Delta \vdash_{\mathrm{a}} U_i' \leq [\overline{V/X}]U_i \qquad \Delta \Vdash_{\mathrm{a}} [\overline{V/X}]\overline{\mathcal{P}} \qquad \Delta \vdash_{\mathrm{a}} \overline{T}, \overline{V}\ \mathsf{ok}\end{array}}{\Delta; \Gamma \vdash_{\mathrm{a}} I\langle \overline{W} \rangle[\overline{T}].m\langle \overline{V} \rangle(\overline{e}) : [\overline{V/X}]U}$$

$$\textsc{exp-alg-new}\quad \frac{(\forall i)\ \Delta; \Gamma \vdash_{\mathrm{a}} e_i : T_i \qquad \Delta \vdash_{\mathrm{a}} N\ \mathsf{ok} \qquad \mathsf{fields}(N) = \overline{U\, f} \qquad (\forall i)\ \Delta \vdash_{\mathrm{a}} T_i \leq U_i}{\Delta; \Gamma \vdash_{\mathrm{a}} \mathtt{new}\, N(\overline{e}) : N}$$

$$\textsc{exp-alg-cast}\quad \frac{\Delta \vdash_{\mathrm{a}} N\ \mathsf{ok} \qquad \Delta; \Gamma \vdash_{\mathrm{a}} e : T}{\Delta; \Gamma \vdash_{\mathrm{a}} (N)\, e : N}$$

Figure 19: Algorithmic expression typing. The relations $\Delta \vdash_{\mathrm{a}} T\ \mathsf{ok}$, $\Delta \vdash_{\mathrm{a}} \mathcal{P}\ \mathsf{ok}$, and $\mathsf{a\text{-}smtype}_{\Delta}(m, K[\overline{T}]) = \langle \overline{X} \rangle\, \overline{U\, x} \rightarrow U\ \mathtt{where}\ \overline{\mathcal{P}}$ are defined as the relations $\Delta \vdash T\ \mathsf{ok}$, $\Delta \vdash \mathcal{P}\ \mathsf{ok}$, and $\mathsf{smtype}_{\Delta}(m, K[\overline{T}]) = \langle \overline{X} \rangle\, \overline{U\, x} \rightarrow U\ \mathtt{where}\ \overline{\mathcal{P}}$, respectively, replacing $\vdash$ with $\vdash_{\mathrm{a}}$ and $\Vdash$ with $\Vdash_{\mathrm{a}}$.

5 Well-formedness Criteria

5.1 Well-formedness Criteria for Type Soundness

5.1.1 Class Definitions

For each class

$$\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\ \{\ \overline{T\,f}^{n}\ \ \overline{m : mdef}^{l}\ \}$$

the following well-formedness criteria must hold:

Wf-Class-1 The field names, including names of inherited fields, are unique. That is, $i \neq j \in [n]$ implies $f_i \neq f_j$ and $\mathsf{fields}(N) = \overline{U\,g}$ implies $\overline{f} \cap \overline{g} = \emptyset$.

Wf-Class-2 The method names $\overline{m}$ are unique. That is, $i \neq j \in [l]$ implies $m_i \neq m_j$.

5.1.2 Interface Definitions

Definition 5.1 ($\mathsf{at\text{-}top}$)**.** *We define* $\mathsf{at\text{-}top}(\overline{X}, T)$ *as* $\overline{X} \cap \mathsf{ftv}(T) = \emptyset$ *or* $T \in \overline{X}$.

For each interface

$$\texttt{interface}\ I\langle\overline{X}\rangle\ [\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\ \{\ \overline{m : \texttt{static}\ msig}\ \ \overline{rcsig}\ \}$$

the following well-formedness criteria must hold:

Wf-Iface-1 The method names $\overline{m}$ are pairwise disjoint.

Wf-Iface-2 In all constraints $\overline{G}\,\texttt{implements}\,K \in \overline{R}$, the implementing types $\overline{Y}$ do not occur in K; that is, $\overline{Y} \cap \mathsf{ftv}(K) = \emptyset$.

Wf-Iface-3 In all constraints $\overline{G}\,\texttt{implements}\,K \in \overline{R}$, the types $\overline{G}$ are pairwise distinct type variables from $\overline{Y}$; that is, $\overline{G} \subseteq \overline{Y}$ and $G_i \neq G_j$ for $i \neq j$.

Wf-Iface-4 In all method signatures $\langle\overline{Z}\rangle\,\overline{T\,x} \to U\ \texttt{where}\ \overline{Q} \in \overline{rcsig}$, the implementing types $\overline{Y}$ may occur only at the top level of $\overline{T}$ and U, and they do not appear in $\overline{Q}$; that is, $\mathsf{at\text{-}top}(\overline{Y}, T_i)$ for all i, $\mathsf{at\text{-}top}(\overline{Y}, U)$, and $\mathsf{ftv}(\overline{Q}) \cap \overline{Y} = \emptyset$.

5.1.3 Implementation Definitions

Definition 5.2 (Dispatch types and positions)**.** *A* dispatch type *of interface I is an implementing type that appears in every non-static method signature of I and all its superinterfaces at the top level of some argument type. The* set of dispatch positions *of I, written* $\mathsf{disp}(I)$*, contains the indices of those implementing types that are dispatch types. See Fig. 20 for a definition of* disp.

For each implementation

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots$$

the following well-formedness criteria must hold:

$$\textsc{disp-iface}\quad \frac{\begin{array}{c}\texttt{interface}\ I\langle\overline{X}\rangle\ [\overline{Y}^{n}\ \texttt{where}\ \overline{R}^{m}]\ \texttt{where}\ \overline{P}\ \{\ \ldots\ \overline{rcsig}^{n}\ \}\\ (\forall i \in [n], i \neq j)\ Y_j \in \mathsf{disp}(rcsig_i) \qquad (\forall i \in [m])\ Y_j \in \mathsf{disp}(R_i)\end{array}}{j \in \mathsf{disp}(I)}$$

$$\textsc{disp-rcsig}\quad \frac{(\forall i)\ Y \in \mathsf{disp}(msig_i)}{Y \in \mathsf{disp}(\texttt{receiver}\,\{\overline{msig}\})}$$

$$\textsc{disp-msig}\quad \frac{Y \notin \overline{X} \qquad Y \in \overline{T}}{Y \in \mathsf{disp}(\langle\overline{X}\rangle\,\overline{T\,x} \to T\ \texttt{where}\ \overline{P})}$$

$$\textsc{disp-constr}\quad \frac{(\forall i)\ \text{if}\ G_i = Y\ \text{then}\ i \in \mathsf{disp}(I)}{Y \in \mathsf{disp}(\overline{G}\,\texttt{implements}\,I\langle\overline{V}\rangle)}$$

Figure 20: Dispatch types and positions.

Wf-Impl-1 The dispatch types among $\overline{N}$ fully determine the type variables $\overline{X}$; that is $\overline{X} \subseteq \mathsf{ftv}(\{N_i \mid i \in \mathsf{disp}(I)\})$.

Wf-Impl-2 There exist suitable implementations for all superinterfaces of I. Suppose $\overline{T}$ **implements** $J\langle\overline{U}\rangle \in \mathsf{sup}(\overline{N}$ **implements** $I\langle\overline{V}\rangle)$. Then there exists a definition $\mathtt{implementation}\langle\overline{Y}\rangle\ J\langle\overline{U'}\rangle\ [\overline{M}]\ \mathtt{where}\ \overline{Q}\ \ldots$ and a substitution $\sigma = [\overline{W/Y}]$ such that

1. $\overline{P} \vdash_{\mathrm{q}} \overline{T} \leq \sigma\overline{M}$ and $T_i = \sigma M_i$ for those i with $i \notin \mathsf{pos}^-(J)$,
2. $\overline{U} = \sigma\overline{U'}$, and
3. $\overline{P} \Vdash_{\mathrm{q}} \sigma\overline{Q}$.

5.1.4 Programs

Definition 5.3 (Greatest lower bound)**.** *The greatest lower bound of G_1 and G_2 with respect to Δ, written $\Delta \vdash G_1 \sqcap G_2$, is defined as follows:*

$$\text{GLB-LEFT}\ \frac{\Delta \vdash G_1 \leq G_2}{\Delta \vdash G_1 \sqcap G_2 = G_1} \qquad \text{GLB-RIGHT}\ \frac{\Delta \vdash G_2 \leq G_1}{\Delta \vdash G_1 \sqcap G_2 = G_2}$$

The notation $\Delta \vdash \overline{G} \sqcap \overline{G'} = \overline{H}$ abbreviates $(\forall i)\ \Delta \vdash G_i \sqcap G'_i = H_i$.

The CoreGI program under consideration must fulfill the following well-formedness criteria:

Wf-Prog-1 Names of non-static interface methods are globally unique.

Wf-Prog-2 A program does not contain two implementations for different instantiations of the same interface or for different non-dispatch types. That is, for each pair of implementation definitions

$$\mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{M}]\ \mathtt{where}\ \overline{P}\ \ldots \qquad \mathtt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\overline{N}]\ \mathtt{where}\ \overline{Q}\ \ldots$$

and for all substitutions $[\overline{V/X}]$ and $[\overline{W/Y}]$ such that $\emptyset \vdash [\overline{V/X}]M_i \sqcap [\overline{W/Y}]N_i$ exists for all $i \in \mathsf{disp}(I)$, it holds that $[\overline{V/X}]\overline{T} = [\overline{W/Y}]\overline{U}$ and that $[\overline{V/X}]M_j = [\overline{W/Y}]N_j$ for all $j \notin \mathsf{disp}(I)$.

Wf-Prog-3 Implementation definitions are downward closed. That is, for each pair of implementation definitions

$$\mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{N}]\ \mathtt{where}\ \overline{P}\ \ldots$$
$$\mathtt{implementation}\langle\overline{X'}\rangle\ I\langle\overline{T'}\rangle\ [\overline{N'}]\ \mathtt{where}\ \overline{P'}\ \ldots$$

and for all substitutions $[\overline{V/X}]$ and $[\overline{V'/X'}]$ with $\emptyset \vdash [\overline{V/X}]\overline{N} \sqcap [\overline{V'/X'}]\overline{N'} = \overline{M}$ there exists an implementation definition

$$\mathtt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\overline{M'}]\ \mathtt{where}\ \overline{Q}\ \ldots$$

and a substitution $[\overline{W/Y}]$ such that $\overline{M} = [\overline{W/Y}]\overline{M'}$.

Wf-Prog-4 Constraints on implementation definitions are consistent with constraints on implementation definitions for subclasses. That is, for each pair of implementation definitions

$$\mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{M}]\ \mathtt{where}\ \overline{P}\ \ldots \qquad \mathtt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\overline{N}]\ \mathtt{where}\ \overline{Q}\ \ldots$$

and for all substitutions $[\overline{V/X}]$ and $[\overline{W/Y}]$ with $[\overline{V/X}]\overline{M} \trianglelefteq_{\mathrm{c}} [\overline{W/Y}]\overline{N}$ and $\emptyset \Vdash [\overline{W/Y}]\overline{Q}$, it holds that $\emptyset \Vdash [\overline{V/X}]\overline{P}$.

5.2 Well-formedness Criteria for Determinacy of Evaluation

5.2.1 Programs

The CoreGI program under consideration must fulfill the following well-formedness criteria:

WF-PROG-5 The class and interface graphs of the program are acyclic. (Each class definition $\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ D\langle\overline{T}\rangle \dots$ contributes an edge $C \to D$ to the class graph, and each interface definition $\texttt{interface}\ I\langle\overline{X}\rangle\ [\overline{Y}\ \texttt{where}\ \overline{P}] \dots$ and each constraint $\overline{G}\ \texttt{implements}\ J\langle\overline{V}\rangle \in \overline{P}$ contributes an edge $I \to J$ to the interface graph.)

WF-PROG-6 A program does not contain two different implementations for the same interface with unifiable implementation types. That is, for each pair of disjoint implementation definitions

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{M}]\ \texttt{where}\ \overline{P} \dots \qquad \texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{Q} \dots$$

it holds that, for all substitutions $[\overline{V/X}]$ and $[\overline{W/Y}]$, $[\overline{V/X}]\overline{M} \neq [\overline{W/Y}]\overline{N}$.

5.3 Well-formedness Criteria for Termination of Entailment and Subtyping Algorithms

5.3.1 Implementation Definitions

For each implementation

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P} \dots$$

the following well-formedness criteria must hold:

WF-IMPL-3 In all constraints $\overline{G}\ \texttt{implements}\ K \in \overline{P}$, the types $\overline{G}$ are type variables from $\overline{X}$; that is, $\overline{G} \subseteq \overline{X}$.

5.3.2 Type Environments

Definition 5.4 (Contractive type environments). *A type environment Δ is* contractive *if there exist no type variables $X_1, \dots, X_n$ such that $X_1 = X_n$ and $X_i\ \texttt{extends}\ X_{i+1} \in \Delta$ for each $i \in \{1, \dots, n-1\}$.*

Definition 5.5 (Closure of types). *The* closure *of a set of types $\mathscr{T}$ with respect to a type environment Δ, written $\mathsf{cls}_\Delta(\mathscr{T})$, is defined as the least set closed under the following rules:*

$$\text{CLS-ID}\ \frac{T \in \mathscr{T}}{T \in \mathsf{cls}_\Delta(\mathscr{T})} \qquad \text{CLS-UP}\ \frac{T \in \mathsf{cls}_\Delta(\mathscr{T}) \qquad \Delta \vdash_{\mathrm{a}}{}' T \leq N}{N \in \mathsf{cls}_\Delta(\mathscr{T})}$$

$$\text{CLS-DECOMP}\ \frac{B\langle\overline{T}\rangle \in \mathsf{cls}_\Delta(\mathscr{T}) \qquad (\textit{where } B = C \textit{ or } B = I)}{T_i \in \mathsf{cls}_\Delta(\mathscr{T})}$$

Each type environment Δ must fulfill the following well-formedness criteria:

WF-TENV-1 The type environment Δ is finite.

WF-TENV-2 The type environment Δ is contractive.

WF-TENV-3 If $\mathscr{T}$ is a finite set of types, then the closure of $\mathscr{T}$ with respect to Δ is finite.

5.4 Well-formedness Criteria for Decidable Expression Typing

5.4.1 Programs

The CoreGI program under consideration must fulfill the following well-formedness criteria:

WF-PROG-7 Multiple instantiation inheritance for interfaces is not allowed. That is, if $K \trianglelefteq_{\mathrm{i}} I\langle\overline{T}\rangle$ and $K \trianglelefteq_{\mathrm{i}} I\langle\overline{U}\rangle$ then $\overline{T} = \overline{U}$.

WF-PROG-8 Multiple inheritance for single-headed interfaces that are neither positive nor negative is not allowed. That is, if $1 \notin \mathsf{pos}^{+}(I)$, $1 \notin \mathsf{pos}^{-}(I)$, $I\langle\overline{T}\rangle \trianglelefteq_{\mathrm{i}} K_1$, and $I\langle\overline{T}\rangle \trianglelefteq_{\mathrm{i}} K_2$, then either $K_1 \trianglelefteq_{\mathrm{i}} K_2$ or $K_2 \trianglelefteq_{\mathrm{i}} K_1$.

5.4.2 Type Environments

Each type environment Δ must fulfill the following well-formedness criteria:

WF-TENV-4 A type variable does not have several unrelated G-types among its bounds. That is, if $X\,\texttt{extends}\,G_1 \in \Delta$ and $X\,\texttt{extends}\,G_2 \in \Delta$ then $\Delta \vdash G_1 \leq G_2$ or $\Delta \vdash G_2 \leq G_1$.

WF-TENV-5 A type variable is not a subtype of different instantiations of the same interface. That is, if $\Delta \vdash_{\mathrm{a}}{}' X \leq I\langle\overline{T}\rangle$ and $\Delta \vdash_{\mathrm{a}}{}' X \leq I\langle\overline{U}\rangle$ then $\overline{T} = \overline{U}$.

WF-TENV-6 A type variable has only negative interfaces among its bounds. That is, if $X\,\texttt{extends}\,I\langle\overline{T}\rangle \in \Delta$ then $1 \in \mathsf{pos}^{-}(I)$.

WF-TENV-7 The type environment Δ does not contain two implementations for different instantiations of the same interface or for different non-dispatch types. That is:

1. For each pair of constraints
$$\begin{array}{c}\overline{G}\,\texttt{implements}\,I\langle\overline{T}\rangle \in \mathsf{sup}(\Delta)\\ \overline{H}\,\texttt{implements}\,I\langle\overline{W}\rangle \in \mathsf{sup}(\Delta)\end{array}$$
such that $\Delta \vdash G_i \sqcap H_i$ exists for all $i \in \mathsf{disp}(I)$, it holds that $\overline{T} = \overline{W}$ and $G_j = H_j$ for all $j \notin \mathsf{disp}(I) \cup \mathsf{pos}^{-}(I)$.
2. For each constraint and each implementation definition
$$\begin{array}{c}\overline{G}\,\texttt{implements}\,I\langle\overline{T}\rangle \in \mathsf{sup}(\Delta)\\ \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{W}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots\end{array}$$
such that $\Delta \vdash G_i \sqcap [\overline{U/X}]N_i$ exists for all $i \in \mathsf{disp}(I)$ and some $\overline{U}$, it holds that $\overline{T} = [\overline{U/X}]\overline{W}$ and $G_j = [\overline{U/X}]N_j$ for all $j \notin \mathsf{disp}(I) \cup \mathsf{pos}^{-}(I)$.

6 Equivalence of Declarative and Quasi-algorithmic Versions of Entailment and Subtyping

We make the global assumption that the underlying program *prog* is well-formed, that is $\vdash$ *prog* ok. Especially, this implies that class, interface, and implementation definitions of the underlying program are closed. Moreover, we assume that types and constraints are only formed from classes and interfaces defined in the underlying program.

Definition 6.1 ($\mathcal{E} \in \mathsf{sup}(\mathcal{E})$)**.** *We generalize the definition of* sup *to* extends*-constraints:*

$$T\,\texttt{extends}\,U \in \mathsf{sup}(T\,\texttt{extends}\,U) \qquad \frac{\vdash K \trianglelefteq_{\mathrm{i}} L}{T\,\texttt{extends}\,L \in \mathsf{sup}(T\,\texttt{extends}\,K)}$$

Lemma 6.2 (Class and interface inheritance is transitive)**.** *If $N_1 \trianglelefteq_{\mathrm{c}} N_2$ and $N_2 \trianglelefteq_{\mathrm{c}} N_3$ then $N_1 \trianglelefteq_{\mathrm{c}} N_3$. If $K_1 \trianglelefteq_{\mathrm{i}} K_2$ and $K_2 \trianglelefteq_{\mathrm{i}} K_3$ then $K_1 \trianglelefteq_{\mathrm{i}} K_3$.*

PROOF. By straightforward inductions on the derivations of $N_1 \trianglelefteq_c N_2$ and $K_1 \trianglelefteq_i K_2$, respectively. $\square$

Lemma 6.3. *If $N \trianglelefteq_c N'$ and $N' \neq$ `Object` then $N \neq$ `Object`.*

PROOF. Follows because programs do not define `Object` explicitly. $\square$

Lemma 6.4 (Kernel of quasi-algorithmic subtyping is reflexive)**.** *$\Delta \vdash_q{}' T \leq T$ is derivable for all types T.*

PROOF. If T is a type variable or an interface type, the claim follows with SUB-Q-ALG-VAR and SUB-Q-ALG-IFACE, respectively. If T is a class type, then the claim follows with SUB-Q-ALG-CLASS, unless $T =$ `Object`, then it follows with SUB-Q-ALG-OBJ. $\square$

Lemma 6.5 (Kernel of quasi-algorithmic subtyping is transitive)**.** *If $\mathcal{D}_1 :: \Delta \vdash_q{}' T \leq U$ and $\mathcal{D}_2 :: \Delta \vdash_q{}' U \leq V$ then $\Delta \vdash_q{}' T \leq V$.*

PROOF. By induction on $\mathcal{D}_1$.
Case distinction on the last rule used in $\mathcal{D}_1$.

- *Case* SUB-Q-ALG-OBJ: Then $\mathcal{D}_2$ also ends with SUB-Q-ALG-OBJ (rule SUB-Q-ALG-CLASS is impossible because of Lemma 6.3). Hence, $V =$ `Object` and the claim follows with SUB-Q-ALG-OBJ.
- *Case* SUB-Q-ALG-VAR-REFL: Trivial because $T = U$.
- *Case* SUB-Q-ALG-VAR: By inverting the rule we get $T = X$, X **extends** $T' \in \Delta$, and $\Delta \vdash_q{}' T' \leq U$. If $V = X$ or $V =$ `Object`, then the claim follows directly. Otherwise, we apply the I.H. to $\Delta \vdash_q{}' T' \leq U$ and get $\Delta \vdash_q{}' T' \leq V$. The claim now follows with SUB-Q-ALG-VAR.
- *Case* SUB-Q-ALG-CLASS: If $V =$ `Object`, then the claim follows with SUB-Q-ALG-OBJ, otherwise by applying Lemma 6.2.
- *Case* SUB-Q-ALG-IFACE: Follows from Lemma 6.2.

End case distinction on the last rule used in $\mathcal{D}_1$. $\square$

Lemma 6.6. *If $\Delta \Vdash_q{}' K\ \texttt{implements}\ L$ then $K \trianglelefteq_i L$.*

PROOF. Assume $K = I\langle\overline{V}\rangle$. Then we have

$$\frac{1 \in \mathsf{pos}^+(I) \qquad I\langle\overline{V}\rangle \trianglelefteq_i L}{\Delta \Vdash_q{}' I\langle\overline{V}\rangle\ \texttt{implements}\ L}\ \text{ENT-Q-ALG-IFACE}$$

so $I\langle\overline{V}\rangle \trianglelefteq_i L$ as required. $\square$

Definition 6.7 ($\in^+$ and $\in^*$)**.**

$$\in^+\text{-DIRECT}\quad \frac{X\ \texttt{extends}\ T \in \Delta}{X\ \texttt{extends}\ T \in^+ \Delta} \qquad \in^+\text{-STEP}\quad \frac{X\ \texttt{extends}\ Y \in \Delta \qquad Y\ \texttt{extends}\ T \in^+ \Delta}{X\ \texttt{extends}\ T \in^+ \Delta} \qquad \in^*\text{-ID}\quad X\ \texttt{extends}\ X \in^* \Delta$$

$$\in^*\text{-PLUS}\quad \frac{X\ \texttt{extends}\ T \in^+ \Delta}{X\ \texttt{extends}\ T \in^* \Delta}$$

Lemma 6.8 (Transitivity of $\in^+$ and $\in^*$)**.**

(*i*) *If $X\ \texttt{extends}\ Y \in^+ \Delta$ and $Y\ \texttt{extends}\ T \in^+ \Delta$ then $X\ \texttt{extends}\ T \in^+ \Delta$.*

(*ii*) *If $X\ \texttt{extends}\ Y \in^* \Delta$ and $Y\ \texttt{extends}\ T \in^* \Delta$ then $X\ \texttt{extends}\ T \in^* \Delta$.*

PROOF. Claim (i) is proved by induction on the derivation of $X\,\texttt{extends}\,Y \in^+ \Delta$. Claim (ii) follows by claim (i) and a case distinction on the last rule used in the derivation of $X\,\texttt{extends}\,Y \in^* \Delta$. □

Lemma 6.9 ($\in^+$ and $\in^*$ imply subtyping)**.** *If $X\,\texttt{extends}\,T \in^+ \Delta$ or $X\,\texttt{extends}\,T \in^* \Delta$ then $\Delta \vdash_{\mathrm{q}}{}' X \leq T$.*

PROOF. If $X\,\texttt{extends}\,T \in^+ \Delta$ then the claim follows by a straightforward induction on the derivation given. The other case is now trivial. Note that we use Lemma 6.4. □

Definition 6.10 (B and $B \trianglelefteq_{\mathrm{ci}} B$)**.** *We let B range over both class types (N) and interface types (K). When we write $B \trianglelefteq_{\mathrm{ci}} B'$, then either $B = N$, $B' = N'$, and $N \trianglelefteq_{\mathrm{c}} N'$, or $B = K$, $B' = K'$, and $K \trianglelefteq_{\mathrm{i}} K'$.*

Lemma 6.11 (Inversion of kernel of quasi-algorithmic subtyping)**.** *Suppose $\Delta \vdash_{\mathrm{q}}{}' T \leq U$.*

(i) If $T = X$ for some X then either $U = Y$ for some Y and $X\,\texttt{extends}\,Y \in^* \Delta$, or $U = \texttt{Object}$, or $U = B$ for some $B \neq \texttt{Object}$ and $X\,\texttt{extends}\,B' \in^+ \Delta$ for some B' with $B' \trianglelefteq_{\mathrm{ci}} B$.

(ii) If $U = Y$ for some Y then $T = X$ for some X and $X\,\texttt{extends}\,Y \in^* \Delta$.

(iii) If $T = N$ for some N then $U = N'$ for some N' with $N \trianglelefteq_{\mathrm{c}} N'$.

(iv) If $T = K$ for some K then either $U = K'$ for some K' with $K \trianglelefteq_{\mathrm{i}} K'$ or $U = \texttt{Object}$.

PROOF. Propositions (3) and (4) follow by inspecting the rules defining the relation $\cdot \vdash_{\mathrm{q}}{}' \cdot \leq \cdot$. Proposition (2) follows by inspecting the rules defining the relation $\cdot \vdash_{\mathrm{q}}{}' \cdot \leq \cdot$ and by proposition (1).

We now prove proposition (1) by induction on the derivation of $\Delta \vdash_{\mathrm{q}}{}' T \leq U$. Thereby, we assume that $U \neq \texttt{Object}$ as the claim holds trivially in this case. Because $T = X$, the derivation either ends with SUB-Q-ALG-VAR-REFL or SUB-Q-ALG-VAR. The first case is trivial. For the second case we have

$$\frac{X\,\texttt{extends}\,U' \in \Delta \qquad U \neq \texttt{Object}, U \neq X \qquad \Delta \vdash_{\mathrm{q}}{}' U' \leq U}{\Delta \vdash_{\mathrm{q}}{}' X \leq U}\ \textsc{sub-q-alg-var}$$

Case distinction on the form of U'.

- *Case* $U' = Z$ for some Z: Applying the I.H. to $\Delta \vdash_{\mathrm{q}}{}' U' \leq U$ yields that either $U = Y$ for some Y and $Z\,\texttt{extends}\,Y \in^* \Delta$, or that $U = B$ for some B and $Z\,\texttt{extends}\,B' \in^+ \Delta$ for some B' with $B' \trianglelefteq_{\mathrm{ci}} B$. It is easy to verify that proposition (1) follows from these facts.
- *Case* $U' = B'$ for some B': Using propositions (3) and (4) we get that $U = B$ for some $B \neq \texttt{Object}$ with $B' \trianglelefteq_{\mathrm{ci}} B$. The claim now follows trivially.

End case distinction on the form of U'. □

Lemma 6.12. *If $\Delta \vdash_{\mathrm{q}}{}' T \leq U$ and $\Delta \vdash_{\mathrm{q}} U \leq V$, then $\Delta \vdash_{\mathrm{q}} T \leq V$.*

PROOF. If the derivation of $\Delta \vdash_{\mathrm{q}} U \leq V$ ends with SUB-Q-ALG-KERNEL, then $\Delta \vdash_{\mathrm{q}}{}' U \leq V$ so the claim follows by Lemma 6.5. Otherwise, we have $V = K$ and

$$\frac{\Delta \vdash_{\mathrm{q}}{}' U \leq U' \qquad \Delta \Vdash_{\mathrm{q}}{}' U'\,\texttt{implements}\,K}{\Delta \vdash_{\mathrm{q}} U \leq K}\ \textsc{sub-q-alg-impl}$$

With Lemma 6.5 we have $\Delta \vdash_{\mathrm{q}}{}' T \leq U'$, so the claim follows with SUB-Q-ALG-IMPL. □

Lemma 6.13 (Type substitution preserves inheritance)**.** *If $\vdash B \trianglelefteq_{\mathrm{ci}} B'$ then $\vdash \sigma B \trianglelefteq_{\mathrm{ci}} \sigma B'$.*

Proof. We show the claim for $B = K$ and $B' = K'$ by induction on the derivation of $K \trianglelefteq_{\mathrm{i}} K'$; the proof for $B = N$ and $B' = N'$ is similar.
Case distinction on the last rule used in $K \trianglelefteq_{\mathrm{i}} K'$.

- *Case* ext-i-refl: Trivial because $K = K'$.
- *Case* ext-i-super: Then $K = I\langle\overline{T}\rangle$ and

$$\frac{\texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \ldots \qquad R_i = \overline{G}\ \texttt{implements}\ L \qquad [\overline{T/X}]L \trianglelefteq_{\mathrm{i}} K'}{I\langle\overline{T}\rangle \trianglelefteq_{\mathrm{i}} K'}$$

  Applying the I.H. to $[\overline{T/X}]L \trianglelefteq_{\mathrm{i}} K'$ yields $\sigma[\overline{T/X}]L \trianglelefteq_{\mathrm{i}} \sigma K'$. Because the definition of I does not contain free type variables, we have $\sigma[\overline{T/X}]L = [\overline{\sigma T/X}]L$. Hence, $\sigma K \trianglelefteq_{\mathrm{i}} \sigma K'$ by ext-i-super.

End case distinction on the last rule used in $K \trianglelefteq_{\mathrm{i}} K'$. □

Lemma 6.14 (sup is transitive)**.** *If* $\mathcal{R}_3 \in \mathsf{sup}(\mathcal{R}_2)$ *and* $\mathcal{R}_2 \in \mathsf{sup}(\mathcal{R}_1)$ *then* $\mathcal{R}_3 \in \mathsf{sup}(\mathcal{R}_1)$.

Proof. The proof is by induction on the height of the derivation of $\mathcal{R}_3 \in \mathsf{sup}(\mathcal{R}_2)$. The case where this derivation ends with rule sup-id is trivial. Now assume that the derivation ends with rule sup-step:

$$\frac{\texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \ldots \qquad \overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(\mathcal{R}_2)}{\underbrace{[\overline{V/X},\overline{U/Y}]R_k}_{=\mathcal{R}_3} \in \mathsf{sup}(\mathcal{R}_2)}$$

Applying the I.H. to $\overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(\mathcal{R}_2)$ yields $\overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(\mathcal{R}_1)$. Applying rule sup-step then gives us $\mathcal{R}_3 \in \mathsf{sup}(\mathcal{R}_1)$. □

Lemma 6.15 (Type substitution preserves sup)**.** *If* $\mathcal{R} \in \mathsf{sup}(\mathcal{S})$ *then* $\sigma\mathcal{R} \in \mathsf{sup}(\sigma\mathcal{S})$.

Proof. The proof is by induction on the derivation of $\mathcal{R} \in \mathsf{sup}(\mathcal{S})$. The claim holds trivially if this derivation ends with rule sup-id. Now suppose the last rule is sup-step:

$$\frac{\texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \ldots \qquad \overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(\mathcal{S})}{\underbrace{[\overline{V/X},\overline{U/Y}]R_k}_{=\mathcal{R}} \in \mathsf{sup}(\mathcal{S})}$$

By the I.H. we have $\sigma(\overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle) \in \mathsf{sup}(\sigma\mathcal{S})$. Thus, by rule sup-step we get $[\overline{\sigma V/X},\overline{\sigma U/Y}]R_k \in \mathsf{sup}(\sigma\mathcal{S})$. The definition of I does not contain free type variables, so $\mathsf{ftv}(R_k) \subseteq \{\overline{X},\overline{Y}\}$. Hence $[\overline{\sigma V/X},\overline{\sigma U/Y}]R_k = \sigma([\overline{V/X},\overline{U/Y}]R_k) = \sigma\mathcal{R}$. □

Lemma 6.16. *If* $\Delta \vdash_{\mathrm{q}} T \leq U$ *and* $U \neq K$ *for any* K *then* $\Delta \vdash_{\mathrm{q}}' T \leq U$.

Proof. Obvious. □

Lemma 6.17. *Suppose* $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$ *for all* $\mathcal{P} \in \mathsf{sup}(\sigma\Delta')$.

(*i*) *If* $X\ \texttt{extends}\ Y \in^* \Delta'$ *then either* $\Delta \vdash_{\mathrm{q}}' \sigma X \leq \sigma Y$ *or* $\sigma Y = K$ *for some* K *such that* $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$ *for all* K' *with* $K \trianglelefteq_{\mathrm{i}} K'$.

(*ii*) *If* $X\ \texttt{extends}\ B \in^+ \Delta'$ *then either* $\Delta \vdash_{\mathrm{q}}' \sigma X \leq \sigma B$ *or* $\sigma B = K$ *for some* K *such that* $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$ *for all* K' *with* $K \trianglelefteq_{\mathrm{i}} K'$.

Proof. We first prove proposition (1) by induction on the derivation of $X\ \texttt{extends}\ Y \in^* \Delta'$. Then we prove proposition (2) by induction on the derivation of $X\ \texttt{extends}\ B \in^+ \Delta'$.

Proof of proposition (1) If $X = Y$ then the claim follows with Lemma 6.4. Otherwise, we have

$$\frac{X\,\texttt{extends}\,Y' \in \Delta' \qquad Y'\,\texttt{extends}\,Y \in^* \Delta'}{X\,\texttt{extends}\,Y \in^* \Delta'}$$

By the assumption we have

$$\Delta \vdash_{\mathrm{q}} \sigma X \le \sigma Y' \tag{1}$$ {eq:subst-x-st-subst-yp::lemma:in-tenv-closed-subst}

and, if $\sigma Y' = L$ for some L, then

$$\Delta \vdash_{\mathrm{q}} \sigma X \le L' \text{ for all } L' \text{ with } L \trianglelefteq_{\mathrm{i}} L' \tag{2}$$ {eq:subst-x-st-lp::lemma:in-tenv-closed-subst}

Applying the I.H. to $Y'\,\texttt{extends}\,Y \in^* \Delta'$ yields either

$$\Delta \vdash_{\mathrm{q}}{}' \sigma Y' \le \sigma Y \tag{3}$$ {eq:subst-yp-st-subst-y::lemma:in-tenv-closed-subst}

or $\sigma Y = K$ for some K and

$$\Delta \vdash_{\mathrm{q}} \sigma Y' \le K' \text{ for all } K' \text{ with } K \trianglelefteq_{\mathrm{i}} K' \tag{4}$$ {eq:subst-yp-st-kp::lemma:in-tenv-closed-subst}

Case distinction on the form of $\sigma Y'$ and on whether (3) or (4) holds.

- *Case* $\sigma Y' \neq L$ for any L and (3) holds: By Lemma 6.16 and (1) we get

  $$\Delta \vdash_{\mathrm{q}}{}' \sigma X \le \sigma Y'$$

  With (3) and Lemma 6.5 we get $\Delta \vdash_{\mathrm{q}}{}' \sigma X \le \sigma Y$ as required.

- *Case* $\sigma Y' \neq L$ for any L and (4) holds: As in the preceding case, we have $\Delta \vdash_{\mathrm{q}}{}' \sigma X \le \sigma Y'$. Using (4) and Lemma 6.12 we get

  $$\Delta \vdash_{\mathrm{q}} \sigma X \le K' \text{ for all } K' \text{ with } K \trianglelefteq_{\mathrm{i}} K'$$

  as required.

- *Case* $\sigma Y' = L$ for some L and (3) holds: With (3) and Lemma 6.11 we get either that $\sigma Y = \texttt{Object}$ or that $\sigma Y = K$ for some K with $L \trianglelefteq_{\mathrm{i}} K$. If $\sigma Y = \texttt{Object}$ then $\Delta \vdash_{\mathrm{q}}{}' \sigma X \le \sigma Y$ by sub-q-alg-obj. Now assume $\sigma Y = K$. With (2) and $L \trianglelefteq_{\mathrm{i}} K$ we get $\Delta \vdash_{\mathrm{q}} \sigma X \le K'$ for all K' with $K \trianglelefteq_{\mathrm{i}} K'$.

- *Case* $\sigma Y' = L$ for some L and (4) holds: Suppose $K \trianglelefteq_{\mathrm{i}} K'$ for some K'.

  – If the derivation of $\Delta \vdash_{\mathrm{q}} \sigma Y' \le K'$ in (4) ends with sub-q-alg-kernel, then we have $\Delta \vdash_{\mathrm{q}}{}' \sigma Y' \le K'$. Hence, by Lemma 6.11: $L \trianglelefteq_{\mathrm{i}} K'$. Using (2) we get $\Delta \vdash_{\mathrm{q}} \sigma X \le K'$.

  – If the derivation of $\Delta \vdash_{\mathrm{q}} \sigma Y' \le K'$ in (4) ends with sub-q-alg-impl, we have

  $$\frac{\Delta \vdash_{\mathrm{q}}{}' \sigma Y' \le T \qquad \Delta \Vdash_{\mathrm{q}}{}' T\,\texttt{implements}\,K'}{\Delta \vdash_{\mathrm{q}} \sigma Y' \le K'}$$

  With Lemma 6.11 we need to consider two cases for the form of T:

  * $T = \texttt{Object}$. Then we have $\Delta \vdash_{\mathrm{q}}{}' \sigma X \le T$, so $\Delta \vdash_{\mathrm{q}} \sigma X \le K'$.
  * $T = L'$ and $L \trianglelefteq_{\mathrm{i}} L'$. With Lemma 6.6 we get $L' \trianglelefteq_{\mathrm{i}} K'$. Thus, $L \trianglelefteq_{\mathrm{i}} K'$. Equation (2) then gives us $\Delta \vdash_{\mathrm{q}} \sigma X \le K'$.

  We now have $\Delta \vdash_{\mathrm{q}} \sigma X \le K'$ for all K' with $K \trianglelefteq_{\mathrm{i}} K'$ as required.

End case distinction on the form of $\sigma Y'$ and on whether (3) or (4) holds.

Proof of proposition (2) We have

$$\frac{X\ \texttt{extends}\ Y \in^* \Delta' \qquad Y\ \texttt{extends}\ B \in \Delta'}{X\ \texttt{extends}\ B \in^+ \Delta'}$$

By proposition (1) we have that either

$$\Delta \vdash_{\mathrm{q}}' \sigma X \leq \sigma Y \tag{5}$$ {eq:pos1::lemma:in-tenv-closed-subst}

or that

$$\sigma Y = L \text{ for some } L \text{ and } \Delta \vdash_{\mathrm{q}} \sigma X \leq L' \text{ for all } L' \text{ with } L \trianglelefteq_{\mathrm{i}} L' \tag{6}$$ {eq:pos2::lemma:in-tenv-closed-subst}

We have by the assumption

$$\Delta \vdash_{\mathrm{q}} \sigma Y \leq \sigma B \tag{7}$$ {eq:ass1::lemma:in-tenv-closed-subst}

and, if $\sigma B = K$ for some K then

$$\Delta \vdash_{\mathrm{q}} \sigma Y \leq K' \text{ for all } K' \text{ with } K \trianglelefteq_{\mathrm{i}} K' \tag{8}$$ {eq:ass2::lemma:in-tenv-closed-subst}

Case distinction on the form of σB and on whether (5) or (6) holds.

- *Case* $\sigma B = N$ for some N and (5) holds: Then by (7) and Lemma 6.16: $\Delta \vdash_{\mathrm{q}}' \sigma Y \leq \sigma B$. With (5) and Lemma 6.5: $\Delta \vdash_{\mathrm{q}}' \sigma X \leq \sigma B$ as required.

- *Case* $\sigma B = K$ for some K and (5) holds: Assume K' such that $K \trianglelefteq_{\mathrm{i}} K'$.

  - If the derivation of $\Delta \vdash_{\mathrm{q}} \sigma Y \leq K'$ in (8) ends with SUB-Q-ALG-KERNEL, then $\Delta \vdash_{\mathrm{q}}' \sigma Y \leq K'$, so $\Delta \vdash_{\mathrm{q}}' \sigma X \leq K'$ by (5) and Lemma 6.5. Hence, $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$
  - If the derivation of $\Delta \vdash_{\mathrm{q}} \sigma Y \leq K'$ in (8) ends with SUB-Q-ALG-IMPL, then we have

    $$\frac{\Delta \vdash_{\mathrm{q}}' \sigma Y \leq T \qquad \Delta \Vdash_{\mathrm{q}}' T\ \texttt{implements}\ K'}{\Delta \vdash_{\mathrm{q}} \sigma Y \leq K'}$$

    By (5) and Lemma 6.5 we then have $\Delta \vdash_{\mathrm{q}}' \sigma X \leq T$, thus $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$.

  We now have $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$ for all K' with $K \trianglelefteq_{\mathrm{i}} K'$ as required.

- *Case* $\sigma B = N$ for some N and (6) holds: Then by (7) and Lemma 6.16: $\Delta \vdash_{\mathrm{q}}' \sigma Y \leq \sigma B$. With (6) we know that $\sigma Y = L$ for some L. Hence, by Lemma 6.11: $\sigma B = \texttt{Object}$. We then have $\Delta \vdash_{\mathrm{q}}' \sigma X \leq \sigma B$ by SUB-Q-ALG-OBJ.

- *Case* $\sigma B = K$ for some K and (6) holds: By (6) we have $\sigma Y = L$ for some L. Assume K' such that $K \trianglelefteq_{\mathrm{i}} K'$.

  - If the derivation of $\Delta \vdash_{\mathrm{q}} \sigma Y \leq K'$ in (8) ends with SUB-Q-ALG-KERNEL, then $\Delta \vdash_{\mathrm{q}}' \sigma Y \leq K'$. Hence, $L \trianglelefteq_{\mathrm{i}} K'$ by Lemma 6.11. Using (6) we then have $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$.
  - If the derivation of $\Delta \vdash_{\mathrm{q}} \sigma Y \leq K'$ in (8) ends with SUB-Q-ALG-IMPL, then we have

    $$\frac{\Delta \vdash_{\mathrm{q}}' \sigma Y \leq T \qquad \Delta \Vdash_{\mathrm{q}}' T\ \texttt{implements}\ K'}{\Delta \vdash_{\mathrm{q}} \sigma Y \leq K'}$$

    With Lemma 6.11 we need to consider two cases for the form of T:

    * $T = \texttt{Object}$. Then we have $\Delta \vdash_{\mathrm{q}}' \sigma X \leq T$, so $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$.
    * $T = L'$ and $L \trianglelefteq_{\mathrm{i}} L'$. With Lemma 6.6 we get $L' \trianglelefteq_{\mathrm{i}} K'$. Thus, $L \trianglelefteq_{\mathrm{i}} K'$. Equation (6) then gives us $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$.

We now have $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$ for all K' with $K \trianglelefteq_{\mathrm{i}} K'$ as required.

End case distinction on the form of σB and on whether (5) or (6) holds. $\square$

Lemma 6.18. *Suppose* $\overline{V}\,\texttt{implements}\,J\langle\overline{W}\rangle \in \mathsf{sup}(\overline{T}\,\texttt{implements}\,I\langle\overline{U}\rangle)$ *and* $\Delta \vdash_{\mathrm{q}}{}' T_i \leq T_i'$ *with* $T_i = T_i'$ *unless* $i \in \mathsf{pos}^-(I)$ *for all* i. *Then there exist* $\overline{V'}$ *such that* $\overline{V'}\,\texttt{implements}\,J\langle\overline{W}\rangle \in \mathsf{sup}(\overline{T'}\,\texttt{implements}\,I\langle\overline{U}\rangle)$ *and* $\Delta \vdash_{\mathrm{q}}{}' V_i \leq V_i'$ *with* $V_i = V_i'$ *unless* $i \in \mathsf{pos}^-(J)$ *for all* i.

Proof. By induction on the derivation of $\overline{V}\,\texttt{implements}\,J\langle\overline{W}\rangle \in \mathsf{sup}(\overline{T}\,\texttt{implements}\,I\langle\overline{U}\rangle)$. If the last rule of this derivation is sup-id, then choose $\overline{V'} = \overline{T'}$ and the claim holds trivially. Now suppose the last rule of the derivation is sup-step:

$$\frac{\texttt{interface}\ I'\langle\overline{X}\rangle\,[\overline{Y}^n\,\texttt{where}\,\overline{R}]\ \ldots \qquad \overline{T''}^n\,\texttt{implements}\,I'\langle\overline{U'}\rangle \in \mathsf{sup}(\overline{T}\,\texttt{implements}\,I\langle\overline{U}\rangle)}{[\overline{U'/X},\overline{T''/Y}]R_k \in \mathsf{sup}(\overline{T}\,\texttt{implements}\,\overline{U})}$$

with

$$[\overline{U'/X},\overline{T''/Y}]R_k = \overline{V}\,\texttt{implements}\,J\langle\overline{W}\rangle$$
$$R_k = \overline{G}^m\,\texttt{implements}\,J\langle\overline{W'}\rangle \tag{9}$$ {eq:def-rk::lemma:transitivity-inplus-instar}

Applying the I.H. to $\overline{T''}^n\,\texttt{implements}\,I'\langle\overline{U'}\rangle \in \mathsf{sup}(\overline{T}\,\texttt{implements}\,I\langle\overline{U}\rangle)$ yields the existence of $\overline{T'''}^n$ such that

$$\overline{T'''}\,\texttt{implements}\,I'\langle\overline{U'}\rangle \in \mathsf{sup}(\overline{T'}\,\texttt{implements}\,I\langle\overline{U}\rangle)$$
$$(\forall j \in [n])\ \Delta \vdash_{\mathrm{q}}{}' T_j'' \leq T_j'''$$
$$(\forall j \in [n])\ T_j'' = T_j''' \text{ or } j \in \mathsf{pos}^-(I')$$

We then have by sup-step

$$[\overline{U'/X},\overline{T'''/Y}]R_k \in \mathsf{sup}(\overline{T'}\,\texttt{implements}\,I\langle\overline{U}\rangle) \tag{10}$$ {eq:rk-in-sup::lemma:transitivity-inplus-instar}

Suppose $j \in [n]$ such that $T_j''' \neq T_j''$. Then we have $j \in \mathsf{pos}^-(I')$. By examining the definition of pos^-, we get $Y_j \in \mathsf{pos}^-(R_k)$. The definition of pos^- now gives us

$$Y_j \notin \mathsf{ftv}(\overline{W'}) \tag{11}$$ {eq:yj-notin-mwp::lemma:transitivity-inplus-instar}
$$(\forall i \in [m])\ (Y_j = G_i \text{ and } i \in \mathsf{pos}^-(J)) \text{ or } Y_j \notin \mathsf{ftv}(G_i) \tag{12}$$ {eq:yj-gi::lemma:transitivity-inplus-instar}

Thus, we have with (11) that

$$[\overline{U'/X},\overline{T'''/Y}]\overline{W'} = [\overline{U'/X},\overline{T''/Y}]\overline{W'} = \overline{W} \tag{13}$$ {eq:eq-w::lemma:transitivity-inplus-instar}

Now define

$$\overline{V'}^m = [\overline{U'/X},\overline{T'''/Y}]\overline{G}$$

Then we have with (9), (10), and (13) that

$$\overline{V'}\,\texttt{implements}\,J\langle\overline{W}\rangle \in \mathsf{sup}(\overline{T'}\,\texttt{implements}\,I\langle\overline{U}\rangle)$$

Suppose $i \in [m]$ and $V_i \neq V_i'$. Then there exists $j \in [n]$ such that $Y_j \in \mathsf{ftv}(G_i)$ and $T_j''' \neq T_j''$. By (12) we then have $Y_j = G_i$ and $i \in \mathsf{pos}^-(J)$. Hence, $V_i = T_j''$ and $V_i' = T_j'''$, so $\Delta \vdash_{\mathrm{q}}{}' V_i \leq V_i'$. $\square$

Lemma 6.19. *If* $\Delta \Vdash_{\mathrm{q}}{}' \mathcal{R}$ *then* $\Delta \Vdash_{\mathrm{q}} \mathcal{R}$.

Proof. Obvious with rule ent-q-alg-up. $\square$

Lemma 6.20. *Suppose J is a single-headed interface such that $1 \in \mathsf{pos}^{\pi}(J)$. If now $J\langle\overline{T}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{U}\rangle$ then also $1 \in \mathsf{pos}^{\pi}(I)$.*

PROOF. Induction on the derivation of $J\langle\overline{T}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{U}\rangle$. If the derivation ends with EXT-I-REFL, then $J\langle\overline{T}\rangle = I\langle\overline{U}\rangle$ and the claim holds trivially. Otherwise, assume

$$\frac{\begin{array}{c}\texttt{interface}\ J\langle\overline{X}\rangle\,[Y\ \texttt{where}\ R]\ \ldots \\ R_i = \overline{G}^n\ \texttt{implements}\ J'\langle\overline{V}\rangle \qquad [\overline{T/X}]J'\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{U}\rangle\end{array}}{J\langle\overline{T}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{U}\rangle}\ \textsc{ext-i-super}$$

By criterion WF-IFACE-3 we have $n = 1$ and $G_1 = Y$. With $1 \in \mathsf{pos}^{\pi}(J)$ we have $Y \in \mathsf{pos}^{\pi}(R_i)$ by POS-IFACE, so $1 \in \mathsf{pos}^{\pi}(J')$ by POS-CONSTR. We can now apply the I.H. to $[\overline{T/X}]J'\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{U}\rangle$ and get $1 \in \mathsf{pos}^{\pi}(I)$ as required. □

Lemma 6.21. *If $\mathcal{D} :: \Delta \Vdash_{\mathrm{q}}{}' \overline{T}^n\ \texttt{implements}\ I\langle\overline{U}\rangle$ and there exists $i \in [n]$ such that $T_i = K$ for some K, then $n = 1$ and $1 \in \mathsf{pos}^{+}(I)$.*

PROOF. *Case distinction* on the last rule used in $\mathcal{D}$.

- *Case* ENT-Q-ALG-ENV: Then $\overline{T}^n\ \texttt{implements}\ I\langle\overline{U}\rangle = R$ for some R, which is impossible if there exists $i \in [n]$ such that $T_i = K$ for some K.
- *Case* ENT-Q-ALG-IMPL: We then have $\overline{T} = \overline{N}$ for some $\overline{N}$. Hence, this case is also impossible.
- *Case* ENT-Q-ALG-IFACE: We then have $n = 1$, $T_1 = J\langle\overline{V}\rangle$ for some $J\langle\overline{V}\rangle$, $1 \in \mathsf{pos}^{+}(J)$, and $J\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{U}\rangle$. By Lemma 6.20 also $1 \in \mathsf{pos}^{+}(I)$.

End case distinction on the last rule used in $\mathcal{D}$. □

Lemma 6.22 (Type substitution preserves quasi-algorithmic subtyping and entailment)**.** *Suppose $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$ for all $\mathcal{P} \in \mathsf{sup}(\sigma\Delta')$.*

(*i*) *If $\mathcal{D}_1 :: \Delta' \vdash_{\mathrm{q}}{}' T \leq U$ then either $\Delta \vdash_{\mathrm{q}}{}' \sigma T \leq \sigma U$ or $\sigma U = K$ for some K and $\Delta \vdash_{\mathrm{q}} \sigma T \leq K'$ for all K' with $K \trianglelefteq_{\mathrm{i}} K'$.*

(*ii*) *If $\mathcal{D}_2 :: \Delta' \vdash_{\mathrm{q}} T \leq U$ then $\Delta \vdash_{\mathrm{q}} \sigma T \leq \sigma U$.*

(*iii*) *If $\mathcal{D}_2 :: \Delta' \Vdash_{\mathrm{q}}{}' \mathcal{R}$ then $\Delta \Vdash_{\mathrm{q}} \sigma\mathcal{R}$.*

(*iv*) *If $\mathcal{D}_4 :: \Delta' \Vdash_{\mathrm{q}} \mathcal{Q}$ then $\Delta \Vdash_{\mathrm{q}} \sigma\mathcal{Q}$.*

PROOF. We proceed by induction on the combined height of $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$, and $\mathcal{D}_4$.

(i) *Case distinction* on the last rule used in $\mathcal{D}_1$.

- *Case* SUB-Q-ALG-OBJ: Trivial.
- *Case* SUB-Q-ALG-VAR-REFL: Follows with Lemma 6.4.
- *Case* SUB-Q-ALG-VAR: We have $T = X$. Thus, by Lemma 6.11, we can distinguish three different cases:
  - $U = Y$ for some Y and $X\ \texttt{extends}\ Y \in^{*} \Delta'$. Then the claim follows with Lemma 6.17.
  - $U = \texttt{Object}$. In this case, $\Delta \vdash_{\mathrm{q}}{}' \sigma T \leq \sigma U$ holds by SUB-Q-ALG-OBJ.
  - $U = B$ for some $B \neq \texttt{Object}$ and $X\ \texttt{extends}\ B' \in^{+} \Delta'$ for some B' with $B' \trianglelefteq_{\mathrm{ci}} B$. Then $\sigma B' \trianglelefteq_{\mathrm{i}} \sigma B$ by Lemma 6.13. By Lemma 6.17, we either have $\Delta \vdash_{\mathrm{q}}{}' \sigma X \leq \sigma B'$ or $\sigma B' = L$ for some L and $\Delta \vdash_{\mathrm{q}} \sigma X \leq L'$ for all L' with $L \trianglelefteq_{\mathrm{i}} L'$.
    - For the first case, we note that $\sigma B' \trianglelefteq_{\mathrm{i}} \sigma B$ implies $\Delta \vdash_{\mathrm{q}}{}' \sigma B' \leq \sigma B$. The claim now follows with Lemma 6.5.

* For the second case, we have with $\sigma B' = L$ for some L that $\sigma B = K$ for some K such that $L \trianglelefteq_{\mathrm{i}} K$. If now $K \trianglelefteq_{\mathrm{i}} K'$ then $L \trianglelefteq_{\mathrm{i}} K'$ (by Lemma 6.2), so $\Delta \vdash_{\mathrm{q}} \sigma X \leq K'$ as required.

- *Case* SUB-Q-ALG-CLASS: Follows with Lemma 6.13.
- *Case* SUB-Q-ALG-IFACE: Follows with Lemma 6.13.

End case distinction on the last rule used in $\mathcal{D}_1$.

(ii) *Case distinction* on the last rule used in $\mathcal{D}_2$.

- *Case* SUB-Q-ALG-KERNEL: We have

$$\frac{\Delta' \vdash_{\mathrm{q}}' T \leq U}{\Delta' \vdash_{\mathrm{q}} T \leq U}$$

By part (i) of the I.H., we have either $\Delta \vdash_{\mathrm{q}}' \sigma T \leq \sigma U$ (which implies $\Delta \vdash_{\mathrm{q}} \sigma T \leq \sigma U$) or $\Delta \vdash_{\mathrm{q}} \sigma T \leq \sigma U$, so the claim holds.

- *Case* SUB-Q-ALG-IMPL: We have $U = I\langle\overline{W}\rangle$ for some $I\langle\overline{W}\rangle$ and

$$\frac{\Delta' \vdash_{\mathrm{q}}' T \leq V \qquad \Delta' \Vdash_{\mathrm{q}}' V \,\texttt{implements}\, I\langle\overline{W}\rangle}{\Delta' \vdash_{\mathrm{q}} T \leq I\langle\overline{W}\rangle}$$

Applying parts (i) and (iii) of the I.H. yields

$$\frac{\Delta \vdash_{\mathrm{q}}' \sigma V \leq V' \qquad \text{if } \sigma V \neq V' \text{ then } 1 \in \mathsf{pos}^{-}(I) \qquad \Delta \Vdash_{\mathrm{q}}' V' \,\texttt{implements}\, \sigma I\langle\overline{W}\rangle}{\Delta \Vdash_{\mathrm{q}} \sigma V \,\texttt{implements}\, \sigma I\langle\overline{W}\rangle}\ \textsc{ent-q-alg-up} \tag{14}$$ {eq:subst-v-impl::lemma:subst-preserves-alg-entailment-subtyping}

and either

$$\Delta \vdash_{\mathrm{q}}' \sigma T \leq \sigma V \tag{15}$$ {eq:pos1::lemma:subst-preserves-alg-entailment-subtyping}

or

$$\sigma V = L \text{ for some } L \text{ and } \Delta \vdash_{\mathrm{q}} \sigma T \leq L' \text{ for all } L' \text{ with } L \trianglelefteq_{\mathrm{i}} L' \tag{16}$$ {eq:pos2::lemma:subst-preserves-alg-entailment-subtyping}

  – Suppose (15). Then we have by the first premise in (14), by (15), and by Lemma 6.12 that $\Delta \vdash_{\mathrm{q}}' \sigma T \leq V'$. With the last premise in (14) and with rule SUB-Q-ALG-IMPL, we then get $\Delta \vdash_{\mathrm{q}} \sigma T \leq \sigma I\langle\overline{W}\rangle$ as required.
  – Suppose (16). Then we have by the first premise in (14), by the fact that $\sigma V = L$, and by Lemma 6.11 that either $V' = \texttt{Object}$ or that $V' = L'$ for some L' with $L \trianglelefteq_{\mathrm{i}} L'$.
    * If $V' = \texttt{Object}$ then $\Delta \vdash_{\mathrm{q}}' \sigma T \leq V'$, so the claim follows with the last premise in (14) and with rule SUB-Q-ALG-IMPL.
    * Otherwise, $V' = L'$ and $L \trianglelefteq_{\mathrm{i}} L'$. From the last premise in (14), we have $\Delta \vdash_{\mathrm{q}}' L' \,\texttt{implements}\, \sigma I\langle\overline{W}\rangle$, so with Lemma 6.6 we get $L' \trianglelefteq_{\mathrm{i}} \sigma I\langle\overline{W}\rangle$. Hence, $L \trianglelefteq_{\mathrm{i}} \sigma I\langle\overline{W}\rangle$ by Lemma 6.2. By (16) we then have $\Delta \vdash_{\mathrm{q}} \sigma T \leq \sigma I\langle\overline{W}\rangle$ as required (note that $\sigma I\langle\overline{W}\rangle = \sigma U$).

End case distinction on the last rule used in $\mathcal{D}_2$.

(iii) *Case distinction* on the last rule used in $\mathcal{D}_3$.

- *Case* ENT-Q-ALG-ENV: We have $S \in \Delta'$ and $R \in \mathsf{sup}(S)$ such that $R = \mathcal{R}$. With Lemma 6.15 we get $\sigma R \in \mathsf{sup}(\sigma S)$. Clearly, $\sigma S \in \sigma\Delta'$, so the assumption gives us $\Delta \Vdash_{\mathrm{q}} \sigma R$ as required.

- *Case* ENT-Q-ALG-IMPL: We have

$$\frac{\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots \qquad \Delta' \Vdash_{\mathrm{q}} [\overline{V/X}]\overline{P}}{\Delta' \Vdash_{\mathrm{q}}{}' \underbrace{[\overline{V/X}](\overline{N}\ \texttt{implements}\ I\langle\overline{T}\rangle)}_{=\mathcal{R}}}$$

Applying part (iv) of the I.H. yields

$$\Delta \Vdash_{\mathrm{q}} \sigma[\overline{V/X}]\overline{P}$$

Because implementation definitions do not contain free type variables, we have

$$\sigma[\overline{V/X}]\overline{P} = [\overline{\sigma V/X}]\overline{P}$$
$$\sigma[\overline{V/X}](\overline{N}\ \texttt{implements}\ I\langle\overline{T}\rangle) = [\overline{\sigma V/X}](\overline{N}\ \texttt{implements}\ I\langle\overline{T}\rangle)$$

By ENT-Q-ALG-IMPL we then have $\Delta \Vdash_{\mathrm{q}}{}' \sigma[\overline{V/X}](\overline{N}\ \texttt{implements}\ I\langle\overline{T}\rangle)$, thus $\Delta \Vdash_{\mathrm{q}} \sigma\mathcal{R}$ by Lemma 6.19.

- *Case* ENT-Q-ALG-IFACE: We have

$$\frac{1 \in \mathsf{pos}^{+}(I) \qquad I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} K}{\Delta' \Vdash_{\mathrm{q}}{}' \underbrace{I\langle\overline{V}\rangle\ \texttt{implements}\ K}_{=\mathcal{R}}}$$

By Lemma 6.13, we have $\sigma I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} \sigma K$. Thus, with ENT-Q-ALG-IFACE, we get $\Delta \Vdash_{\mathrm{q}}{}' \sigma\mathcal{R}$, so $\Delta \Vdash_{\mathrm{q}} \sigma\mathcal{R}$ by Lemma 6.19.

End case distinction on the last rule used in $\mathcal{D}_3$.

(iv) *Case distinction* on the last rule used in $\mathcal{D}_4$.

- *Case* ENT-Q-ALG-EXTENDS: Follows from part (ii) of the I.H.
- *Case* ENT-Q-ALG-UP: We have

$$\frac{(\forall i)\ \Delta' \vdash_{\mathrm{q}}{}' T_i \leq U_i \quad \text{if } T_i \neq U_i \text{ then } i \in \mathsf{pos}^{-}(I) \qquad \Delta' \Vdash_{\mathrm{q}}{}' \overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle}{\Delta' \Vdash_{\mathrm{q}} \underbrace{\overline{T}^{n}\ \texttt{implements}\ I\langle\overline{V}\rangle}_{=\mathcal{Q}}} \tag{17}$$ {eq:ass::lemma:subst-preserves-alg-entailment-subtyping}

We get by part (iii) of the I.H.:

$$\frac{\begin{array}{c}(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' \sigma U_i \leq U'_i \quad \text{if } \sigma U_i \neq U'_i \text{ then } i \in \mathsf{pos}^{-}(I)\\ \Delta \Vdash_{\mathrm{q}}{}' \overline{U'}\ \texttt{implements}\ I\langle\overline{\sigma V}\rangle\end{array}}{\Delta \Vdash_{\mathrm{q}} \sigma\overline{U}\ \texttt{implements}\ I\langle\overline{\sigma V}\rangle}\ \text{ENT-Q-ALG-UP} \tag{18}$$ {eq:subst-mu-ent::lemma:subst-preserves-alg-entailment-subtyping}

Suppose $i \in [n]$. If $i \in \mathsf{pos}^{-}(I)$ does not hold, then we have $T_i = U_i$ and $\sigma U_i = U'_i$. Hence,

$$\sigma T_i = U'_i \text{ or } i \in \mathsf{pos}^{-}(I) \tag{19}$$ {eq:ti-eq-uip::lemma:subst-preserves-alg-entailment-subtyping}

Moreover, by part (i) of the I.H. applied to the first premise in (17) we get that either

$$\Delta \vdash_{\mathrm{q}}{}' \sigma T_i \leq \sigma U_i \tag{20}$$ {eq:pos21::lemma:subst-preserves-alg-entailment-subtyping}

or

$$\sigma U_i = K_i \text{ for some } K_i \text{ and } \Delta \vdash_{\mathrm{q}} \sigma T_i \leq K'_i \text{ for all } K'_i \text{ with } K_i \trianglelefteq_{\mathrm{i}} K'_i \tag{21}$$ {eq:pos22::lemma:subst-preserves-alg-entailment-subtyping}

We now partition $[n] = \mathscr{M}_1 \,\dot\cup\, \mathscr{M}_2$ such that

$$\mathscr{M}_1 = \{j \in [n] \mid \text{Equation (20) holds for } j\}$$
$$\mathscr{M}_2 = \{l \in [n] \mid \text{Equation (21) holds for } l\}$$

– If $j \in \mathscr{M}_1$, then we have with (20), the first premise in (18), and Lemma 6.5 that $\Delta \vdash_{\mathrm{q}}{}' \sigma T_j \leq U'_j$.
– If $l \in \mathscr{M}_2$, then $\sigma U_l = K_l$ for some K_l. By Lemma 6.11 applied to the first premise in (18), we then have that either $U'_l = K'_l$ for some K'_l or $U'_l = \texttt{Object}$.

Now we further partition $\mathscr{M}_2$ into $\mathscr{M}_{21} \dot\cup \mathscr{M}_{22}$ such that

$$\mathscr{M}_{21} = \{l \in \mathscr{M}_2 \mid U'_l = K'_l \text{ for some } K'_l\}$$
$$\mathscr{M}_{22} = \{l \in \mathscr{M}_2 \mid U'_l = \texttt{Object}\}$$

Case distinction on whether or not $\mathscr{M}_{21} = \emptyset$.

– *Case* $\mathscr{M}_{21} = \emptyset$: Then we have $[n] = \mathscr{M}_1 \dot\cup \mathscr{M}_{22}$, so $\Delta \vdash_{\mathrm{q}}{}' \sigma T_i \leq U'_i$ for all $i \in [n]$. Thus, with (19) and the last premise in (18) we can apply ENT-Q-ALG-UP and get $\Delta \Vdash_{\mathrm{q}} \sigma\overline{T}\ \texttt{implements}\ I\langle\overline{\sigma V}\rangle$ as required.
– *Case* $\mathscr{M}_{21} \neq \emptyset$: With Lemma 6.21 applied to the last premise in (18), we get that $n = 1$ and that

$$1 \in \mathsf{pos}^+(I) \tag{22}$$ {eq:pos-plus::lemma:subst-preserves-alg-entailment-subtyping}

In the following, we may assume

$$1 \in \mathsf{pos}^-(I) \tag{23}$$ {eq:pos-minus::lemma:subst-preserves-alg-entailment-subtyping}

Otherwise, we have $\sigma T_1 = U'_1$ with (19) and the claim then follows with the last premise in (18) and ENT-Q-ALG-UP.
With $n = 1$ and $\mathscr{M}_{21} \neq \emptyset$, we have $1 \in \mathscr{M}_{21}$. Hence, $U'_1 = K'_1$ for some K'_1. With the last premise in (18) and Lemma 6.6 we then have $K'_1 \trianglelefteq_{\mathrm{i}} I\langle\overline{\sigma V}\rangle$. Because $1 \in \mathscr{M}_{21} \subseteq \mathscr{M}_2$, we have we have $\sigma U_1 = K_1$ for some K_1. The first premise in (18) and Lemma 6.11 then gives us $K_1 \trianglelefteq_{\mathrm{i}} K'_1$. With Lemma 6.2: $K_1 \trianglelefteq_{\mathrm{i}} I\langle\overline{\sigma V}\rangle$. Equation (21) holds because $1 \in \mathscr{M}_2$, so

$$\Delta \vdash_{\mathrm{q}} \sigma T_1 \leq I\langle\overline{\sigma V}\rangle \tag{24}$$ {eq:subst-t1::lemma:subst-preserves-alg-entailment-subtyping}

Case distinction on the last rule used in the derivation of (24).

∗ *Case* SUB-Q-ALG-KERNEL: Then $\Delta \vdash_{\mathrm{q}}{}' \sigma T_1 \leq I\langle\overline{\sigma V}\rangle$. With (22) and (23) we then have

$$\dfrac{1 \in \mathsf{pos}^-(I) \qquad \dfrac{\Delta \vdash_{\mathrm{q}}{}' \sigma T_1 \leq I\langle\overline{\sigma V}\rangle \qquad \dfrac{1 \in \mathsf{pos}^+(I) \qquad I\langle\overline{\sigma V}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{\sigma V}\rangle}{\Delta \Vdash_{\mathrm{q}}{}' I\langle\overline{\sigma V}\rangle\ \texttt{implements}\ I\langle\overline{\sigma V}\rangle}\ \textsc{ent-q-alg-iface}}{}}{\Delta \Vdash_{\mathrm{q}} \sigma T_1\ \texttt{implements}\ I\langle\overline{\sigma V}\rangle}\ \textsc{ent-q-alg-up}$$

∗ *Case* SUB-Q-ALG-IMPL: We then have

$$\dfrac{\Delta \vdash_{\mathrm{q}}{}' \sigma T_1 \leq W \qquad \Delta \Vdash_{\mathrm{q}}{}' W\ \texttt{implements}\ I\langle\overline{\sigma V}\rangle}{\Delta \vdash_{\mathrm{q}} \sigma T_1 \leq I\langle\overline{\sigma V}\rangle}$$

With (23) we get

$$\dfrac{\Delta \vdash_{\mathrm{q}}{}' \sigma T_1 \leq W \qquad 1 \in \mathsf{pos}^-(I) \qquad \Delta \Vdash_{\mathrm{q}}{}' W\ \texttt{implements}\ I\langle\overline{\sigma V}\rangle}{\Delta \Vdash_{\mathrm{q}} \sigma T_1\ \texttt{implements}\ I\langle\overline{\sigma V}\rangle}\ \textsc{ent-q-alg-up}$$

End case distinction on the last rule used in the derivation of (24).

End case distinction on whether or not $\mathscr{M}_{21} = \emptyset$.

We thus showed $\Delta \Vdash_{\mathrm{q}} \sigma\mathcal{Q}$.

End case distinction on the last rule used in $\mathcal{D}_4$. $\square$

Lemma 6.23. *If* $\mathcal{R} \in \mathsf{sup}(T\,\mathtt{implements}\,L)$ *then* $\mathcal{R} = T\,\mathtt{implements}\,L'$ *with* $L \trianglelefteq_{\mathrm{i}} L'$.

PROOF. We proceed by induction on the derivation of $\mathcal{R} \in \mathsf{sup}(T\,\mathtt{implements}\,L)$.
Case distinction on the last rule of the derivation of $\mathcal{R} \in \mathsf{sup}(T\,\mathtt{implements}\,L)$.

- *Case* rule SUP-ID: Obvious.
- *Case* rule SUP-STEP: We have

$$\frac{\mathtt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \mathtt{where}\,\overline{S}]\ \ldots \qquad \overline{U}\,\mathtt{implements}\,I\langle\overline{V}\rangle \in \mathsf{sup}(T\,\mathtt{implements}\,L)}{[\overline{V/X},\overline{U/Y}]S_j \in \mathsf{sup}(T\,\mathtt{implements}\,L)}$$

with $\mathcal{R} = [\overline{V/X},\overline{U/Y}]S_j$. Applying the I.H. yields

$$\overline{U}\,\mathtt{implements}\,I\langle\overline{V}\rangle = T\,\mathtt{implements}\,I\langle\overline{V}\rangle$$
$$L \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle$$

Hence,

$$\overline{Y} = Y$$

By criterion WF-IFACE-2 and criterion WF-IFACE-3, we have

$$S_j = Y\,\mathtt{implements}\,K$$
$$Y \notin \mathsf{ftv}(K)$$

Hence,

$$[\overline{V/X},\overline{U/Y}]S_j = T\,\mathtt{implements}\,[\overline{V/X}]K$$

Moreover,

$$I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} [\overline{V/X}]K$$

Hence, with Lemma 6.2

$$L \trianglelefteq_{\mathrm{i}} [\overline{V/X}]K$$

End case distinction on the last rule of the derivation of $\mathcal{R} \in \mathsf{sup}(T\,\mathtt{implements}\,L)$. $\square$

Lemma 6.24. *If* $\mathcal{S} \in \mathsf{sup}(R)$ *then there exists a* S *with* $\mathcal{S} = S$.

PROOF. By induction on the derivation of $\mathcal{S} \in \mathsf{sup}(R)$. The case where the derivation ends with rule SUP-ID is trivial because $\mathcal{S} = R$. Now suppose that the derivation ends with an application of rule SUP-STEP:

$$\frac{\mathtt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \mathtt{where}\,\overline{S}]\ \ldots \qquad \overline{U}\,\mathtt{implements}\,I\langle\overline{V}\rangle \in \mathsf{sup}(R)}{\underbrace{[\overline{V/X},\overline{U/Y}]S_k}_{=\mathcal{S}} \in \mathsf{sup}(R)}$$

Suppose $S_k = \overline{G}\,\mathtt{implements}\,K$. By using the I.H., we get that there exists $\overline{H}$ such that $\overline{U} = \overline{H}$. From criterion WF-IFACE-3 we then know that $\{[\overline{V/X},\overline{U/Y}]\overline{G}\} \subseteq \{\overline{H}\}$. Thus, there exists $S = \mathcal{S}$. $\square$

Lemma 6.25. *If* $\mathcal{R} \in \mathsf{sup}(\sigma\mathcal{S})$ *then there exists a* $\mathcal{R}' \in \mathsf{sup}(\mathcal{S})$ *with* $\sigma\mathcal{R}' = \mathcal{R}$.

Proof. By induction on the derivation of $\mathcal{R} \in \mathsf{sup}(\sigma\mathcal{S})$. The case where the derivation ends with rule SUP-ID is trivial because $\mathcal{R} = \sigma\mathcal{S}$. Now suppose that the derivation ends with an application of rule SUP-STEP:

$$\frac{\texttt{interface}\ I\langle\overline{X}\rangle\ [\overline{Y\ \texttt{where}\ R}]\ \ldots \qquad \overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(\sigma\mathcal{S})}{\underbrace{[\overline{V/X},\overline{U/Y}]R_k}_{=\mathcal{R}} \in \mathsf{sup}(\sigma\mathcal{S})}$$

From the I.H. we get the existence of $\overline{U'}$ and $\overline{V'}$ such that $\overline{U'}\ \texttt{implements}\ I\langle\overline{V'}\rangle \in \mathsf{sup}(\mathcal{S})$ and $\sigma\overline{U'} = \overline{U}$, $\sigma\overline{V'} = \overline{V}$. By rule SUP-STEP we then have

$$[\overline{V'/X},\overline{U'/Y}]Q_k \in \mathsf{sup}(\mathcal{S})$$

Define $\mathcal{R}' = [\overline{V'/X},\overline{U'/Y}]R_k$. We then get

$$\sigma\mathcal{R}' = \sigma[\overline{V'/X},\overline{U'/Y}]R_k \overset{\mathsf{ftv}(R_k)\subseteq\{\overline{X},\overline{Y}\}}{=} [\overline{\sigma V'/X},\overline{\sigma U'/Y}]R_k = [\overline{V/X},\overline{U/Y}]R_k = \mathcal{R}$$

as required. □

Lemma 6.26. *Suppose I is a single-headed interface. If $I\langle\overline{T}\rangle \trianglelefteq_{\mathsf{i}} K$, then $U\ \texttt{implements}\ K \in \mathsf{sup}(U\ \texttt{implements}\ I\langle\overline{T}\rangle)$ for any U.*

Proof. By induction on the derivation of $I\langle\overline{T}\rangle \trianglelefteq_{\mathsf{i}} K$. If the derivation ends with EXT-I-REFL, then $I\langle\overline{T}\rangle = K$ and the claim follows trivially. Now suppose the derivation ends with EXT-I-SUPER:

$$\frac{\texttt{interface}\ I\langle\overline{X}\rangle\ [Y\ \texttt{where}\ \overline{R}]\ \ldots \qquad R_i = \overline{G}\ \texttt{implements}\ L \qquad [\overline{T/X}]L \trianglelefteq_{\mathsf{i}} K}{I\langle\overline{T}\rangle \trianglelefteq_{\mathsf{i}} K}$$

By applying the I.H. to $[\overline{T/X}]L \trianglelefteq_{\mathsf{i}} K$, we get

$$U\ \texttt{implements}\ K \in \mathsf{sup}(U\ \texttt{implements}\ [\overline{T/X}]L) \tag{25}$$ {eq:1::lemma:extends-iface-sup}

for any type U.

By SUP-ID, we have $U\ \texttt{implements}\ I\langle\overline{T}\rangle \in \mathsf{sup}(U\ \texttt{implements}\ I\langle\overline{T}\rangle)$. Thus, by SUP-STEP also $[U/Y,\overline{T/X}]R_i \in \mathsf{sup}(U\ \texttt{implements}\ I\langle\overline{T}\rangle)$ With criterion WF-IFACE-3 we have $\overline{G} = Y$ and with criterion WF-IFACE-2 $Y \notin \mathsf{ftv}(L)$. Thus, $[U/Y,\overline{T/X}]R_i = U\ \texttt{implements}\ [\overline{T/X}]L$. Hence,

$$U\ \texttt{implements}\ [\overline{T/X}]L \in \mathsf{sup}(U\ \texttt{implements}\ I\langle\overline{T}\rangle) \tag{26}$$ {eq:2::lemma:extends-iface-sup}

With Lemma 6.14 we then get $U\ \texttt{implements}\ K \in \mathsf{sup}(U\ \texttt{implements}\ I\langle\overline{T}\rangle)$ as required. □

Lemma 6.27 (Inversion of quasi-algorithmic entailment)**.** *If $\Delta \Vdash_{\mathsf{q}} \overline{T}^n\ \texttt{implements}\ I\langle\overline{V}\rangle$ then there exist $\overline{U}^n$ such that $\Delta \Vdash_{\mathsf{q}}{}'\ \overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle$, and for all $i \in [n]$, $\Delta \vdash_{\mathsf{q}}{}'\ T_i \leq U_i$ and $i \in \mathsf{pos}^-(I)$ unless $T_i = U_i$.*

Proof. The derivation of $\Delta \Vdash_{\mathsf{q}} \overline{T}^n\ \texttt{implements}\ I\langle\overline{V}\rangle$ must end with rule ENT-Q-ALG-UP. The claim now follows from the premises of this rule. □

Lemma 6.28 (Entailment for super constraints)**.**

(i) If $\mathcal{D}_1 :: \Delta \Vdash_{\mathsf{q}} \mathcal{P}$ and $\mathcal{Q} \in \mathsf{sup}(\mathcal{P})$, then $\Delta \Vdash_{\mathsf{q}} \mathcal{Q}$.

(ii) If $\mathcal{D}_2 :: \Delta \Vdash_{\mathsf{q}}{}'\ \mathcal{R}$ and $\mathcal{S} \in \mathsf{sup}(\mathcal{R})$, then $\Delta \Vdash_{\mathsf{q}} \mathcal{S}$.

(iii) If $\mathcal{D}_3 :: \Delta \vdash_{\mathsf{q}} T \leq K$ and $K \trianglelefteq_{\mathsf{i}} L$, then $\Delta \vdash_{\mathsf{q}} T \leq L$.

Proof. We proceed by induction on the combined height of $\mathcal{D}_1$ and $\mathcal{D}_2$.

(i) *Case distinction* on the last rule used in $\mathcal{D}_1$.

- *Case* ENT-Q-ALG-EXTENDS: Then

$$\frac{\Delta \vdash_{\mathrm{q}} T \le U}{\Delta \Vdash_{\mathrm{q}} \underbrace{T\,\texttt{extends}\,U}_{=\mathcal{P}}}$$

If U is not an interface type, the $\mathcal{Q} = \mathcal{P}$ and the claim holds trivially. Otherwise $U = K$ for some K and $\mathcal{Q} = T\,\texttt{extends}\,L$ for some L with $K \trianglelefteq_{\mathrm{i}} L$. By part (iii) of the I.H., we get $\Delta \vdash_{\mathrm{q}} T \le L$. Hence, $\Delta \Vdash_{\mathrm{q}} \mathcal{Q}$ by ENT-Q-ALG-EXTENDS.

- *Case* ENT-Q-ALG-UP: Then we have

$$\frac{(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' T_i \le T_i' \qquad \text{if } T_i \neq T_i' \text{ then } i \in \mathsf{pos}^-(I) \qquad \Delta \Vdash_{\mathrm{q}}{}' \overline{T'}\,\texttt{implements}\,I\langle\overline{U}\rangle}{\Delta \Vdash_{\mathrm{q}} \underbrace{\overline{T}\,\texttt{implements}\,I\langle\overline{U}\rangle}_{=\mathcal{P}}} \tag{27}$$ {eq:entails-p::lemma:sup-implies-entailment-subtyping}

Assume $\mathcal{Q} = \overline{V}\,\texttt{implements}\,J\langle\overline{W}\rangle$. With Lemma 6.18 we get the existence of $\overline{V'}$ such that

$$\overline{V'}\,\texttt{implements}\,J\langle\overline{W}\rangle \in \mathsf{sup}(\overline{T'}\,\texttt{implements}\,I\langle\overline{U}\rangle) \tag{28}$$ {eq:sup-p::lemma:sup-implies-entailment-subtyping}

$$(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' V_i \le V_i' \tag{29}$$ {eq:vi-st-vip::lemma:sup-implies-entailment-subtyping}

$$(\forall i) \text{ if } V_i \neq V_i' \text{ then } i \in \mathsf{pos}^-(J) \tag{30}$$ {eq:pos-minus::lemma:sup-implies-entailment-subtyping}

Applying part (ii) of the I.H. to (28) and the last premise in (27) yields

$$\Delta \Vdash_{\mathrm{q}} \overline{V'}\,\texttt{implements}\,J\langle\overline{W}\rangle$$

Hence

$$\frac{\begin{array}{c}(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' V_i' \le V_i'' \\ (\forall i) \text{ if } V_i' \neq V_i'' \text{ then } i \in \mathsf{pos}^-(J) \qquad \Delta \Vdash_{\mathrm{q}}{}' \overline{V''}\,\texttt{implements}\,J\langle\overline{W}\rangle\end{array}}{\Delta \Vdash_{\mathrm{q}} \overline{V'}\,\texttt{implements}\,J\langle\overline{W}\rangle}\ \text{ENT-Q-ALG-UP}$$

With (29) and Lemma 6.5 we get $\Delta \vdash_{\mathrm{q}}{}' V_i \le V_i''$ for all i. Moreover, if $V_i \neq V_i''$ then either $V_i \neq V_i'$ or $V_i' \neq V_i''$; hence, noting (30), we have $i \in \mathsf{pos}^-(J)$ for those i with $V_i \neq V_i''$. By rule ENT-Q-ALG-UP we then get $\Delta \Vdash_{\mathrm{q}} \overline{V}\,\texttt{implements}\,J\langle\overline{W}\rangle$ as required.

End case distinction on the last rule used in $\mathcal{D}_1$.

(ii) *Case distinction* on the last rule used in $\mathcal{D}_2$.

- *Case* ENT-Q-ALG-ENV: Then $\mathcal{R} = R$ for some R and $R' \in \Delta$ and $R \in \mathsf{sup}(R')$. With Lemma 6.24 we know that there exists $S = \mathcal{S}$. Thus, we also have $S \in \mathsf{sup}(R)$. With Lemma 6.14 we then get $S \in \mathsf{sup}(R')$. Hence, $\Delta \Vdash_{\mathrm{q}}{}' \mathcal{S}$.

- *Case* ENT-Q-ALG-IMPL: We have

$$\frac{\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{Q}\ \ldots \qquad \Delta \Vdash_{\mathrm{q}} \sigma\overline{Q} \qquad \mathsf{dom}(\sigma) = \overline{X}}{\Delta \Vdash_{\mathrm{q}} \underbrace{\sigma(\overline{N}\,\texttt{implements}\,I\langle\overline{V}\rangle)}_{=\mathcal{R}}} \tag{31}$$ {eq:entails-impl::lemma:sup-implies-entailment-subtyping}

From Lemma 6.25 we know that there exists $\mathcal{S}' \in \mathsf{sup}(\overline{N}\,\texttt{implements}\,I\langle\overline{V}\rangle)$ such that $\sigma\mathcal{S}' = \mathcal{S}$. Let $\mathcal{S}' = \overline{T}\,\texttt{implements}\,J\langle\overline{U}\rangle$. By criterion WF-IMPL-2 we get the existence of a definition

$$\texttt{implementation}\langle\overline{Y}\rangle\ J\langle\overline{U'}\rangle\ [\,\overline{M}\,]\ \texttt{where}\ \overline{P}\ \ldots$$

and a substitution $\tau = [\overline{W/Y}]$ such that

$$(\forall i)\ \overline{Q} \vdash_{\mathrm{q}} \overline{T}_i \leq \tau \overline{M}_i \tag{32}$$ {eq:ti-st-subst-mi::lemma:sup-implies-entailment-subtyping}

$$(\forall i) \text{ if } T_i \neq \tau M_i \text{ then } i \in \mathsf{pos}^-(J)$$

$$\overline{U} = \tau \overline{U'} \tag{33}$$ {eq:mu-eq-mup::lemma:sup-implies-entailment-subtyping}

$$\overline{Q} \Vdash_{\mathrm{q}} \tau \overline{P} \tag{34}$$ {eq:entails-subst-mc::lemma:sup-implies-entailment-subtyping}

Applying part (i) of the I.H. to $\Delta \Vdash_{\mathrm{q}} \sigma\overline{Q}$ in (31) yields

$$\Delta \Vdash_{\mathrm{q}} \mathcal{Q}' \text{ for all } \mathcal{Q}' \in \mathsf{sup}(\sigma\overline{Q})$$

Using this equation together with Lemma 6.22, (32), and (34) yields (note that $\sigma\tau M_i \neq K$ for any K)

$$(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' \sigma T_i \leq \sigma\tau M_i \tag{35}$$ {eq:ti-st-subst-substa-mi::lemma:sup-implies-entailment-subtyping}

$$\Delta \Vdash_{\mathrm{q}} \sigma\tau\overline{P} \tag{36}$$ {eq:entails-subst-substa-mc::lemma:sup-implies-entailment-subtyping}

Define σ' with $\mathsf{dom}(\sigma') = \overline{Y}$ and $\sigma'(X) = \sigma\tau(X)$ for all $X \in \overline{Y}$. We then have

$$\dfrac{(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' \sigma T_i \leq \sigma' M_i \text{ with (35)} \qquad (\forall i) \text{ if } \sigma T_i \neq \sigma' M_i \text{ then } i \in \mathsf{pos}^-(J) \qquad \dfrac{\begin{array}{c}\texttt{implementation}\langle\overline{Y}\rangle\ J\langle\overline{U'}\rangle\ [\,\overline{M}\,]\ \texttt{where}\ \overline{P}\ \ldots \\ \Delta \Vdash_{\mathrm{q}} \sigma'\overline{P} \text{ with (36)}\end{array}}{\Delta \Vdash_{\mathrm{q}}{}' \sigma'(\overline{M}\ \texttt{implements}\ J\langle\overline{U'}\rangle)}\ \textsc{ent-q-alg-impl}}{\Delta \Vdash_{\mathrm{q}} \sigma\overline{T}\ \texttt{implements}\ J\langle\overline{\sigma' U'}\rangle}\ \textsc{ent-q-alg-up}$$

With (33) we get $\sigma\overline{U} = \sigma'\overline{U'}$. Hence,

$$\sigma\overline{T}\ \texttt{implements}\ J\langle\overline{\sigma' U'}\rangle = \sigma\overline{T}\ \texttt{implements}\ J\langle\overline{\sigma U}\rangle = \sigma\mathcal{S}' = \mathcal{S}$$

Thus, $\Delta \Vdash_{\mathrm{q}} \mathcal{S}$ as required.

- *Case* ENT-Q-ALG-IFACE: We have

$$\dfrac{1 \in \mathsf{pos}^+(I) \qquad I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} K}{\Delta \Vdash_{\mathrm{q}}{}' \underbrace{I\langle\overline{V}\rangle\ \texttt{implements}\ K}_{=\mathcal{R}}}$$

With Lemma 6.23 we get $\mathcal{S} = I\langle\overline{V}\rangle\ \texttt{implements}\ L$ with $K \trianglelefteq_{\mathrm{i}} L$. With Lemma 6.2 we get $I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} L$. Hence, with ENT-Q-ALG-IFACE, we have $\Delta \Vdash_{\mathrm{q}}{}' \mathcal{S}$.

End case distinction on the last rule used in $\mathcal{D}_2$.

(iii) *Case distinction* on the last rule used in $\mathcal{D}_3$.

- *Case* SUB-Q-ALG-KERNEL: Then we have $\Delta \vdash_{\mathrm{q}}{}' T \leq K$ and from $K \trianglelefteq_{\mathrm{i}} L$ we get $\Delta \vdash_{\mathrm{q}}{}' K \leq L$. Using Lemma 6.5 we get $\Delta \vdash_{\mathrm{q}}{}' T \leq L$, from which we get $\Delta \vdash_{\mathrm{q}} T \leq L$ by rule SUB-Q-ALG-KERNEL.
- *Case* SUB-Q-ALG-IMPL: We have

$$\dfrac{\Delta \vdash_{\mathrm{q}}{}' T \leq U \qquad \Delta \Vdash_{\mathrm{q}}{}' U\ \texttt{implements}\ K}{\Delta \vdash_{\mathrm{q}} T \leq K}$$

With $K \trianglelefteq_{\mathrm{i}} L$ and Lemma 6.26, we get $U\ \texttt{implements}\ L \in \mathsf{sup}(U\ \texttt{implements}\ K$. Thus, with part (ii) of the I.H. we get

$$\Delta \Vdash_{\mathrm{q}} U\ \texttt{implements}\ L$$

By Lemma 6.27 we get the existence of U' such that

$$\Delta \vdash_{\mathrm{q}}{}' U \leq U'$$
$$\Delta \Vdash_{\mathrm{q}}{}' U' \,\texttt{implements}\, L$$

By Lemma 6.5 we then get $\Delta \vdash_{\mathrm{q}}{}' T \leq U'$, so the claim follows by using rule ENT-Q-ALG-IMPL.

End case distinction on the last rule used in $\mathcal{D}_3$. □

Corollary 6.29. *Suppose* $\Delta \Vdash_{\mathrm{q}} \sigma\Delta'$.

(*i*) *If* $\Delta' \vdash_{\mathrm{q}} T \leq U$ *then* $\Delta \vdash_{\mathrm{q}} \sigma T \leq \sigma U$.

(*ii*) *If* $\Delta' \Vdash_{\mathrm{q}} \mathcal{P}$ *then* $\Delta \Vdash_{\mathrm{q}} \sigma\mathcal{P}$.

PROOF. Combine Lemma 6.22 and Lemma 6.28. □

Lemma 6.30 (Transitivity of quasi-algorithmic subtyping). *If* $\mathcal{D}_1 :: \Delta \vdash_{\mathrm{q}} T \leq U$ *and* $\mathcal{D}_2 :: \Delta \vdash_{\mathrm{q}} U \leq V$ *then* $\Delta \vdash_{\mathrm{q}} T \leq V$.

PROOF. *Case distinction* on the last rules used in the derivations $\mathcal{D}_1$ and $\mathcal{D}_2$.

- *Case* SUB-Q-ALG-KERNEL and SUB-Q-ALG-KERNEL: Then the claim follows with Lemma 6.5.
- *Case* SUB-Q-ALG-KERNEL and SUB-Q-ALG-IMPL: Then the claim follows with Lemma 6.12.
- *Case* SUB-Q-ALG-IMPL and SUB-Q-ALG-KERNEL: Then we have $U = K$ for some K. With Lemma 6.11 we get that either $V = \texttt{Object}$ or $V = L$ for some L with $K \trianglelefteq_{\mathrm{i}} L$.

  If $V = \texttt{Object}$, then the claim follows with SUB-Q-ALG-OBJ and SUB-Q-ALG-KERNEL. Otherwise, $V = L$ for some L with $K \trianglelefteq_{\mathrm{i}} L$. The claim now follows with Lemma 6.28.
- *Case* SUB-Q-ALG-IMPL and SUB-Q-ALG-IMPL: We then have $U = K$ for some K and $V = L$ for some L. Moreover,

  $$\frac{\Delta \vdash_{\mathrm{q}}{}' T \leq T' \qquad \Delta \Vdash_{\mathrm{q}}{}' T' \,\texttt{implements}\, K}{\Delta \vdash_{\mathrm{q}} T \leq K} \qquad \frac{\Delta \vdash_{\mathrm{q}}{}' K \leq U' \qquad \Delta \Vdash_{\mathrm{q}}{}' U' \,\texttt{implements}\, L}{\Delta \vdash_{\mathrm{q}} K \leq L}$$

  With $\Delta \vdash_{\mathrm{q}}{}' K \leq U'$ and Lemma 6.11 we know that either $U' = \texttt{Object}$ or $U' = K'$ for some K' with $K \trianglelefteq_{\mathrm{i}} K'$. If $U' = \texttt{Object}$, then $\Delta \vdash_{\mathrm{q}}{}' T \leq U'$ by SUB-Q-ALG-OBJ, so $\Delta \vdash_{\mathrm{q}} T \leq L$ follows by SUB-Q-ALG-IMPL.

  Now suppose $U' = K'$ for some K' with $K \trianglelefteq_{\mathrm{i}} K'$. By Lemma 6.6 and $\Delta \Vdash_{\mathrm{q}}{}' U' \,\texttt{implements}\, L$ we have $K' \trianglelefteq_{\mathrm{i}} L$. Hence, with Lemma 6.2: $K \trianglelefteq_{\mathrm{i}} L$. With $\Delta \vdash_{\mathrm{q}} T \leq K$ and Lemma 6.28 we then get $\Delta \vdash_{\mathrm{q}} T \leq L$ as required.

End case distinction on the last rules used in the derivations $\mathcal{D}_1$ and $\mathcal{D}_2$. □

Lemma 6.31. *If* $\Delta \Vdash_{\mathrm{q}} \overline{T}^{n-1}\, U'\, \overline{V} \,\texttt{implements}\, I\langle\overline{W}\rangle$ *and* $n \in \mathsf{pos}^-(I)$ *and* $\Delta \vdash_{\mathrm{q}}{}' U \leq U'$, *then* $\Delta \Vdash_{\mathrm{q}} \overline{T}^{n-1}\, U\, \overline{V} \,\texttt{implements}\, I\langle\overline{W}\rangle$.

PROOF. From $\Delta \Vdash_{\mathrm{q}} \overline{T}^{n-1}\, U'\, \overline{V} \,\texttt{implements}\, I\langle\overline{W}\rangle$ we get with Lemma 6.27 the existence of $\overline{T'}^{n-1}\, U''\, \overline{V'}$ such that

$$(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' T_i \leq T'_i$$
$$(\forall i)\ \text{if } T_i \neq T'_i \text{ then } i \in \mathsf{pos}^-(I)$$
$$(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' V_i \leq V'_i$$
$$(\forall i)\ \text{if } V_i \neq V'_i \text{ then } n + i \in \mathsf{pos}^-(I)$$
$$\Delta \vdash_{\mathrm{q}}{}' U' \leq U''$$
$$\Delta \Vdash_{\mathrm{q}}{}' \overline{T'}^{n-1}\, U''\, \overline{V'} \,\texttt{implements}\, I\langle\overline{W}\rangle$$

With Lemma 6.5 we then have

$$\Delta \vdash_{\mathrm{q}}{}' U \leq U''$$

Because $n \in \mathsf{pos}^-(I)$ we can apply rule ENT-Q-ALG-UP and get $\Delta \Vdash_{\mathrm{q}} \overline{T}^{n-1}\, U\, \overline{V}\ \texttt{implements}\ I\langle\overline{W}\rangle$ as required. □

Lemma 6.32. *Suppose $\Delta \Vdash_{\mathrm{q}} \overline{T}^n\ \texttt{implements}\ I\langle\overline{W}\rangle$ and $[n] = \mathcal{N}_1 \dot\cup \mathcal{N}_2$ such that $T_i = K_i$ for all $i \in \mathcal{N}_1$ and $T_i = G_i$ for all $i \in \mathcal{N}_2$. Then one of the following holds:*

- *$\Delta \Vdash_{\mathrm{q}} \overline{U}^n\ \texttt{implements}\ I\langle\overline{W}\rangle$ for any $\overline{U}$ with $U_i = G_i$ for all $i \in \mathcal{N}_2$. Moreover, $i \in \mathsf{pos}^-(I)$ for all $i \in \mathcal{N}_1$.*
- *$\mathcal{N}_1 = \{1\}$, $\mathcal{N}_2 = \emptyset$, $1 \in \mathsf{pos}^+(I)$, and $K_1 \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$. Moreover, if $1 \notin \mathsf{pos}^-(I)$ then, if $K_1 = J\langle\overline{W'}\rangle$, $1 \in \mathsf{pos}^+(J)$*

PROOF. From $\Delta \Vdash_{\mathrm{q}} \overline{T}^n\ \texttt{implements}\ I\langle\overline{W}\rangle$ we get with Lemma 6.27 the existence of $\overline{T'}^n$ such that

$$(\forall i \in [n])\ \Delta \vdash_{\mathrm{q}}{}' T_i \leq T_i'$$

$$(\forall i \in [n])\ \text{if } T_i \neq T_i' \text{ then } i \in \mathsf{pos}^-(I) \tag{37}$$ {eq:pos-minus::lemma:alg-entail-iface}

$$\Delta \Vdash_{\mathrm{q}}{}' \overline{T'}^n\ \texttt{implements}\ I\langle\overline{W}\rangle \tag{38}$$ {eq:entails::lemma:alg-entail-iface}

With Lemma 6.11 we know for all $i \in \mathcal{N}_1$ that either $T_i' = K_i'$ for some K_i' with $K_i \trianglelefteq_{\mathrm{i}} K_i'$ or $T_i' = \texttt{Object}$.

- Assume there exists some $i \in \mathcal{N}_1$ such that $T_i' = K_i'$ for some K_i'. Then the derivation of $\Delta \Vdash_{\mathrm{q}}{}' \overline{T'}^n\ \texttt{implements}\ I\langle\overline{W}\rangle$ must end with rule ENT-Q-ALG-IFACE. Hence:

$$[n] = \{1\}$$
$$\mathcal{N}_1 = \{1\}$$
$$\mathcal{N}_2 = \emptyset$$
$$T_1' = J\langle\overline{W'}\rangle\ (= K_1')$$
$$J\langle\overline{W'}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$$
$$1 \in \mathsf{pos}^+(J)$$

  With $K_1 \trianglelefteq_{\mathrm{i}} K_1'$ we then also have $K_1 \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$. With Lemma 6.20: $1 \in \mathsf{pos}^+(I)$.

- Assume $T_i' = \texttt{Object}$ for all $i \in \mathcal{N}_1$. Because $T_i = K_i \neq \texttt{Object}$ we have $i \in \mathsf{pos}^-(I)$ by (37). Let $\overline{U}^n$ be given with $U_i = G_i$ for all $i \in \mathcal{N}_2$. Then

$$(\forall i \in [n])\ \Delta \vdash_{\mathrm{q}}{}' U_i \leq T_i'$$
$$(\forall i \in [n])\ \text{if } U_i \neq T_i' \text{ then } i \in \mathsf{pos}^-(I)$$

  With (38) and rule ENT-Q-ALG-UP we then have $\Delta \Vdash_{\mathrm{q}} \overline{U}^n\ \texttt{implements}\ I\langle\overline{W}\rangle$.

  Finally, suppose $1 \notin \mathsf{pos}^-(I)$. Then $T_1 = T_1'$ by (37), so $K_1 = K_1' = J\langle\overline{W'}\rangle$ and $1 \in \mathsf{pos}^+(J)$ as required. □

Lemma 6.33. *If $\Delta \vdash_{\mathrm{q}}{}' T \leq U$ and $\Delta \Vdash_{\mathrm{q}}{}' U\ \texttt{implements}\ K$ and $K \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle$ and $1 \in \mathsf{pos}^-(I)$, then $\Delta \Vdash_{\mathrm{q}} T\ \texttt{implements}\ I\langle\overline{V}\rangle$.*

PROOF. With $K \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle$ and Lemma 6.26 we have $U\ \texttt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(U\ \texttt{implements}\ K)$. Hence, with Lemma 6.28 we have

$$\Delta \Vdash_{\mathrm{q}} U\ \texttt{implements}\ I\langle\overline{V}\rangle$$

By Lemma 6.27 we then get the existence of U' with

$$\Delta \vdash_{\mathrm{q}}{}' U \leq U'$$
$$\Delta \Vdash_{\mathrm{q}}{}' U' \,\texttt{implements}\, I\langle\overline{V}\rangle$$

By Lemma 6.5 we have $\Delta \vdash_{\mathrm{q}}{}' T \leq U'$, so with $1 \in \mathsf{pos}^{-}(I)$ and rule ENT-Q-ALG-UP, we get $\Delta \Vdash_{\mathrm{q}} T \,\texttt{implements}\, I\langle\overline{V}\rangle$. □

Theorem 6.34 (Completeness of quasi-algorithmic entailment and subtyping)**.**

(*i*) *If* $\Delta \Vdash \mathcal{P}$ *then* $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$.

(*ii*) *If* $\Delta \vdash T \leq U$ *then* $\Delta \vdash_{\mathrm{q}} T \leq U$.

PROOF. Induction on the combined height of the derivations of $\Delta \Vdash \mathcal{P}$ and $\Delta \vdash T \leq U$.

(i) *Case distinction* on the last rule used in the derivation of $\Delta \Vdash \mathcal{P}$.

- *Case* ENT-EXTENDS: Follows with part (ii) of the I.H.
- *Case* ENT-ENV: With rules SUP-ID and ENT-Q-ALG-ENV we have $\Delta \Vdash_{\mathrm{q}}{}' \mathcal{P}$. The claim then follows from Lemma 6.19.
- *Case* ENT-SUPER: Then we have

$$\frac{\texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \ldots \qquad \Delta \Vdash \overline{U}\ \texttt{implements}\ I\langle\overline{T}\rangle}{\Delta \Vdash \underbrace{[\overline{T/X}, \overline{U/Y}]R_i}_{=\mathcal{P}}}$$

  We then get by part (i) of the I.H.

$$\Delta \Vdash_{\mathrm{q}} \overline{U}\ \texttt{implements}\ I\langle\overline{T}\rangle$$

  By looking at the rules defining sup, we also have

$$\mathcal{P} \in \mathsf{sup}(\overline{U}\ \texttt{implements}\ I\langle\overline{T}\rangle)$$

  The claim $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$ now follows from Lemma 6.28.
- *Case* ENT-IMPL: We have

$$\frac{\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots \qquad \Delta \Vdash [\overline{U/X}]\overline{P}}{\Delta \Vdash \underbrace{[\overline{U/X}](\overline{N}\ \texttt{implements}\ I\langle\overline{T}\rangle)}_{=\mathcal{P}}}$$

  By part (i) of the I.H. we get $\Delta \Vdash_{\mathrm{q}} [\overline{U/X}]\overline{P}$. With rule ENT-Q-ALG-IMPL we then have $\Delta \Vdash_{\mathrm{q}}{}' \mathcal{P}$. The claim now follows with Lemma 6.19.
- *Case* ENT-UP: We have

$$\frac{\Delta \vdash U \leq U' \qquad \Delta \Vdash \overline{T}\,U'\,\overline{V}\ \texttt{implements}\ I\langle\overline{W}\rangle \qquad n \in \mathsf{pos}^{-}(I)}{\Delta \Vdash \underbrace{\overline{T}^{n-1}\,U\,\overline{V}^{m}\ \texttt{implements}\ I\langle\overline{W}\rangle}_{=\mathcal{P}}} \tag{39}$$ {eq:init::theorem:completeness-alg-entailment-subtyping}

  Applying part (i) of the I.H. yields

$$\Delta \Vdash_{\mathrm{q}} \overline{T}\,U'\,\overline{V}\ \texttt{implements}\ I\langle\overline{W}\rangle \tag{40}$$ {eq:up-entails::theorem:completeness-alg-entailment-subtyping}

  and part (ii) yields

$$\Delta \vdash_{\mathrm{q}} U \leq U' \tag{41}$$ {eq:u-st-up::theorem:completeness-alg-entailment-subtyping}

  Case distinction on the last rule used in the derivation of (41).

- *Case* rule SUB-Q-ALG-KERNEL: Then $\Delta \vdash_{\mathrm{q}}{}' U \leq U'$. With Lemma 6.31 we then have $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$ as required.
- *Case* rule SUB-Q-ALG-IMPL: Then we have $U' = K$ for some K such that

$$\frac{\Delta \vdash_{\mathrm{q}}{}' U \leq U'' \qquad \Delta \Vdash_{\mathrm{q}}{}' U'' \,\mathtt{implements}\, K}{\Delta \vdash_{\mathrm{q}} U \leq K}$$

Applying Lemma 6.32 to (40) with $U' = K$ yields that either $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$ (we are done in this case) or that $n = 1$, $m = 0$, and $K \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$. With $\Delta \vdash_{\mathrm{q}}{}' U \leq U''$, $\Delta \Vdash_{\mathrm{q}}{}' U'' \,\mathtt{implements}\, K$, $K \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$, $1 \in \mathsf{pos}^{-}(I)$ (follows from (39)), and Lemma 6.33 we get

$$\Delta \Vdash_{\mathrm{q}} U \,\mathtt{implements}\, I\langle\overline{W}\rangle$$

as required.

End case distinction on the last rule used in the derivation of (41).

- *Case* ENT-IFACE: We then have $\mathcal{P} = I\langle\overline{T}\rangle \,\mathtt{implements}\, I\langle\overline{T}\rangle$ and $1 \in \mathsf{pos}^{+}(I)$, so the claim follows with rule ENT-Q-ALG-IFACE.

End case distinction on the last rule used in the derivation of $\Delta \Vdash \mathcal{P}$.

Case distinction on the last rule used in the derivation of $\Delta \vdash T \leq U$.

- *Case* SUB-REFL: Follows with Lemma 6.4 and rule SUB-Q-ALG-KERNEL.
- *Case* SUB-OBJECT: Follows with rules SUB-Q-ALG-OBJ and SUB-Q-ALG-KERNEL.
- *Case* SUB-TRANS: Follows with Lemma 6.30.
- *Case* SUB-VAR: Then we have $T = X$ for some X and $X \,\mathtt{extends}\, U \in \Delta$. If $U = X$ or $U = \mathtt{Object}$, then the claim follows using rules SUB-Q-ALG-VAR-REFL or SUB-Q-ALG-OBJ, respectively, together with SUB-Q-ALG-KERNEL. Otherwise, rule SUB-Q-ALG-VAR is applicable (note Lemma 6.4), so the claim follows with SUB-Q-ALG-KERNEL.
- *Case* SUB-CLASS: Follows with SUB-Q-ALG-CLASS or SUB-Q-ALG-OBJ.
- *Case* SUB-IFACE: Follows with SUB-Q-ALG-IFACE.
- *Case* SUB-IMPL: Then

$$\frac{\Delta \Vdash T \,\mathtt{implements}\, K}{\Delta \vdash T \leq \underbrace{K}_{=U}}$$

Applying the I.H. yields $\Delta \Vdash_{\mathrm{q}} T \,\mathtt{implements}\, K$. With Lemma 6.27 we get the existence of T' such that

$$\Delta \vdash_{\mathrm{q}}{}' T \leq T'$$
$$\Delta \Vdash_{\mathrm{q}}{}' T' \,\mathtt{implements}\, K$$

Using rule SUB-Q-ALG-IMPL we now can derive $\Delta \vdash T \leq K$.

End case distinction on the last rule used in the derivation of $\Delta \vdash T \leq U$. □

Lemma 6.35. *If $\Delta \Vdash \mathcal{R}$ and $\mathcal{S} \in \mathsf{sup}(\mathcal{R})$ then $\Delta \Vdash \mathcal{S}$.*

PROOF. By an induction on the derivation of $\mathcal{S} \in \mathsf{sup}(\mathcal{R})$. If the derivation ends with SUP-ID, then $\mathcal{S} = \mathcal{R}$ and the claim follows trivially. Suppose the derivation ends with SUP-STEP:

$$\frac{\mathtt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \mathtt{where}\ \overline{S}]\ \ldots \qquad \overline{V}\,\mathtt{implements}\, I\langle\overline{W}\rangle \in \mathsf{sup}(\mathcal{R})}{\underbrace{[\overline{W/X}, \overline{V/Y}]S_i}_{=\mathcal{S}} \in \mathsf{sup}(\mathcal{R})}$$

Applying the I.H. yields

$$\Delta \Vdash \overline{V}\,\texttt{implements}\,I\langle\overline{W}\rangle$$

The claim now follows with ENT-SUPER. □

Theorem 6.36 (Soundness of quasi-algorithmic subtyping and entailment)**.**

(*i*) *If* $\mathcal{D}_1 :: \Delta \vdash_{\mathrm{q}}{}' T \leq U$ *then* $\Delta \vdash T \leq U$.

(*ii*) *If* $\mathcal{D}_2 :: \Delta \vdash_{\mathrm{q}} T \leq U$ *then* $\Delta \vdash T \leq U$.

(*iii*) *If* $\mathcal{D}_3 :: \Delta \Vdash_{\mathrm{q}}{}' \mathcal{R}$ *then* $\Delta \Vdash \mathcal{R}$.

(*iv*) *If* $\mathcal{D}_4 :: \Delta \Vdash_{\mathrm{q}} \mathcal{P}$ *then* $\Delta \Vdash \mathcal{P}$.

PROOF. We proceed by induction on the combined height of $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$, and $\mathcal{D}_4$.

(i) *Case distinction* on the last rule used in $\mathcal{D}_1$.

- *Case* SUB-Q-ALG-OBJ: Follows with SUB-OBJECT.
- *Case* SUB-Q-ALG-VAR-REFL: Follows with SUB-REFL.
- *Case* SUB-Q-ALG-VAR: Follows by appeal to part (i) of the I.H., SUB-VAR, and SUB-TRANS.
- *Case* SUB-Q-ALG-CLASS: Follows by combining (possibly repeated) applications of SUB-CLASS with SUB-TRANS.
- *Case* SUB-Q-ALG-IFACE: Follows by combining (possibly repeated) applications of SUB-IFACE with SUB-TRANS.

End case distinction on the last rule used in $\mathcal{D}_1$.

(ii) *Case distinction* on the last rule used in $\mathcal{D}_2$.

- *Case* SUB-Q-ALG-KERNEL: Follows from part (i) of the I.H.
- *Case* SUB-Q-ALG-IMPL: We have

$$\frac{\Delta \vdash_{\mathrm{q}}{}' T \leq T' \qquad \Delta \Vdash_{\mathrm{q}}{}' T'\,\texttt{implements}\,K}{\Delta \vdash_{\mathrm{q}} T \leq \underbrace{K}_{=U}}$$

By parts (i) and (iii) we get

$$\Delta \vdash T \leq T'$$
$$\Delta \Vdash T'\,\texttt{implements}\,K$$

With SUB-IMPL we then have $\Delta \vdash T' \leq K$, so SUB-TRANS yields the desired result.

End case distinction on the last rule used in $\mathcal{D}_2$.

(iii) *Case distinction* on the last rule used in $\mathcal{D}_3$.

- *Case* ENT-Q-ALG-ENV: We then have $S \in \Delta$ and $\mathcal{R} \in \mathsf{sup}(S)$ With ENT-ENV we get $\Delta \Vdash S$. Applying Lemma 6.35 then yields $\Delta \Vdash \mathcal{R}$.
- *Case* ENT-Q-ALG-IMPL: By appeal to part (iv) of the I.H. and rule ENT-IMPL.
- *Case* ENT-Q-ALG-IFACE: We then have $\mathcal{R} = I\langle\overline{V}\rangle\,\texttt{implements}\,K$, $1 \in \mathsf{pos}^{+}(I)$, and $I\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} K$. With Lemma 6.26 we get

$$I\langle\overline{V}\rangle\,\texttt{implements}\,K \in \mathsf{sup}(I\langle\overline{V}\rangle\,\texttt{implements}\,I\langle\overline{V}\rangle)$$

Because $1 \in \mathsf{pos}^{+}(I)$ we have with ENT-IFACE

$$\Delta \Vdash I\langle\overline{V}\rangle\,\texttt{implements}\,I\langle\overline{V}\rangle$$

Then $\Delta \Vdash \mathcal{R}$ by Lemma 6.35.

End case distinction on the last rule used in $\mathcal{D}_3$.

(iv) *Case distinction* on the last rule used in $\mathcal{D}_4$.

- *Case* ENT-Q-ALG-EXTENDS: Follows by part (ii) of the I.H. and ENT-EXTENDS.
- *Case* ENT-Q-ALG-UP: We have $\mathcal{P} = \overline{T}^n\ \texttt{implements}\ I\langle\overline{V}\rangle$ and

$$\frac{(\forall i)\ \Delta \vdash_{\mathrm{q}}{}'\ T_i \leq U_i \qquad (\forall i) \text{ if } T_i \neq U_i \text{ then } i \in \mathsf{pos}^-(I) \qquad \Delta \Vdash_{\mathrm{q}}{}'\ \overline{U}\ \texttt{implements}\ I\langle\overline{V}\rangle}{\Delta \Vdash_{\mathrm{q}} \overline{T}^n\ \texttt{implements}\ I\langle\overline{V}\rangle} \tag{42}$$ {eq:init::lemma:soundness-alg-subtyping-entailment}

By part (i) and (iii), we get

$$(\forall i)\ \Delta \vdash T_i \leq U_i \tag{43}$$ {eq:by-ih1::lemma:soundness-alg-subtyping-entailment}

$$\Delta \Vdash \overline{U}^n\ \texttt{implements}\ I\langle\overline{V}\rangle$$

We now show $\Delta \Vdash \overline{T}^n\ \texttt{implements}\ I\langle\overline{V}\rangle$ by an inner induction on the number m of indices i with $T_i \neq U_i$.

  - If $m = 0$ then $\overline{T} = \overline{U}$ and the claim follows trivially.
  - Assume $m > 0$. W.l.o.g., suppose $T_n \neq U_n$. We get by the inner I.H.

$$\Delta \Vdash \overline{T}^{n-1}\ U_n\ \texttt{implements}\ I\langle\overline{V}\rangle \tag{44}$$ {eq:by-iih::lemma:soundness-alg-subtyping-entailment}

  Because $T_n \neq U_n$ we have $n \in \mathsf{pos}^-(I)$ by (42). With (43),(44), and ENT-UP we then get $\Delta \Vdash \overline{T}^n\ \texttt{implements}\ I\langle\overline{V}\rangle$ as required.

End case distinction on the last rule used in $\mathcal{D}_4$. □

7 Type Soundness

Lemma 7.1 (Type substitution preserves entailment and subtyping)**.** *Suppose* $\Delta \Vdash \sigma\Delta'$.

(*i*) *If* $\Delta' \vdash T \leq U$ *then* $\Delta \vdash \sigma T \leq \sigma U$.

(*ii*) *If* $\Delta' \Vdash \mathcal{P}$ *then* $\Delta \Vdash \sigma\mathcal{P}$.

PROOF. Follows with Corollary 6.29, Theorem 6.34, and Theorem 6.36. □

Lemma 7.2 (Weakening)**.** *Assume* $\Delta \subseteq \Delta'$.

(*i*) *If* $\Delta \Vdash \mathcal{P}$ *then* $\Delta' \Vdash \mathcal{P}$.

(*ii*) *If* $\Delta \vdash T \leq U$ *then* $\Delta' \vdash T \leq U$.

(*iii*) *If* $\Delta \vdash \mathcal{P}\ \mathsf{ok}$ *then* $\Delta' \vdash \mathcal{P}\ \mathsf{ok}$.

(*iv*) *If* $\Delta \vdash T\ \mathsf{ok}$ *then* $\Delta' \vdash T\ \mathsf{ok}$.

PROOF. We prove the first two parts by induction on the combined height of the derivations of $\Delta \Vdash \mathcal{P}$ and $\Delta \vdash T \leq U$. Similarly, we prove the last two parts by induction on the combined height of the derivations of $\Delta \vdash \mathcal{P}\ \mathsf{ok}$ and $\Delta \vdash T\ \mathsf{ok}$. □

Lemma 7.3 (Substitution preserves well-formedness)**.** *Suppose* $\Delta \Vdash \sigma\Delta'$ *and* $\Delta \vdash \sigma X\ \mathsf{ok}$ *for all* $X \in \mathsf{dom}(\sigma)$ *and* $\mathsf{dom}(\Delta) \supseteq \mathsf{dom}(\Delta') \setminus \mathsf{dom}(\sigma)$.

(*i*) *If* $\Delta' \vdash T\ \mathsf{ok}$ *then* $\Delta \vdash \sigma T\ \mathsf{ok}$

(*ii*) *If* $\Delta' \vdash \mathcal{P}\ \mathsf{ok}$ *then* $\Delta \vdash \sigma\mathcal{P}\ \mathsf{ok}$

PROOF. We proceed by induction on the combined height of the two derivations given.

(i) *Case distinction* on the last rule used in the derivation of $\Delta' \vdash T\ \mathsf{ok}$.

- *Case* rule OK-TVAR: Then $T = X$ and $X \in \mathsf{dom}(\Delta')$.
  - If $X \in \mathsf{dom}(\sigma)$ then $\Delta \vdash \sigma X\ \mathsf{ok}$ by assumption.
  - If $X \notin \mathsf{dom}(\sigma)$ then $X \in \mathsf{dom}(\Delta)$ by assumption. Hence, $\Delta \vdash \sigma X\ \mathsf{ok}$.
- *Case* rule OK-OBJECT: Trivial.
- *Case* rule OK-CLASS: Follows from the I.H., Lemma 7.1, and the assumption that classes of the underlying program are closed.
- *Case* rule OK-IFACE: Then

$$\frac{\Delta' \vdash \overline{T}\ \mathsf{ok} \qquad \begin{array}{l}\texttt{interface}\ I\langle\overline{X}\rangle\,[Y\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\ \ldots\\ Y \notin \mathsf{ftv}(\overline{T}, \Delta')\end{array} \qquad \Delta', Y\ \texttt{implements}\ I\langle\overline{T}\rangle \Vdash \overline{[T/X]}\overline{R}, \overline{P}}{\Delta' \vdash I\langle\overline{T}\rangle\ \mathsf{ok}}$$

  with $T = I\langle\overline{T}\rangle$. By the I.H. we have $\Delta \vdash \sigma\overline{T}\ \mathsf{ok}$. W.l.o.g., $Y \notin \mathsf{ftv}(\sigma\overline{T}, \Delta) \cup \mathsf{dom}(\sigma)$. We get with the assumption $\Delta \Vdash \sigma\Delta'$, Lemma 7.2, and rule ENT-ENV that $\Delta, Y\ \texttt{implements}\ I\langle\sigma\overline{T}\rangle \Vdash \sigma(\Delta', Y\ \texttt{implements}\ I\langle\overline{T}\rangle)$. Lemma 7.1 now yields

$$\Delta, Y\ \texttt{implements}\ I\langle\sigma\overline{T}\rangle \Vdash \underbrace{\sigma\overline{[T/X]}\overline{R}, \overline{P}}_{=\overline{[\sigma T/X]}\overline{R}, \overline{P}}$$

  Hence, by rule OK-IFACE, $\Delta \vdash \sigma I\langle\overline{T}\rangle\ \mathsf{ok}$.

End case distinction on the last rule used in the derivation of $\Delta' \vdash T\ \mathsf{ok}$.

(ii) We proceed by case distinction on the last rule used in the derivation of $\Delta' \vdash \mathcal{P}\ \mathsf{ok}$. For rule OK-IMPL-CONSTR, the claim follows with Lemma 7.1 and the I.H. For rules OK-EXT-CONSTR and OK-MONO-CONSTR, the claim follows directly from the I.H. □

Lemma 7.4. *If $\Delta \Vdash \sigma\Delta'$ and $\mathsf{mtype}_{\Delta'}(m, T) = msig$ then $\mathsf{mtype}_{\Delta}(m, \sigma T) = \sigma\, msig$.*

PROOF. Follows by case distinction on the rule used to derive $\mathsf{mtype}_{\Delta'}(m, T) = msig$. The case where this rule is MTYPE-IFACE relies on Lemma 7.1. Moreover, we use the assumption that classes and interfaces of the underlying program are closed. □

Lemma 7.5. *If $\Delta \Vdash \sigma\Delta'$ and $\mathsf{smtype}_{\Delta'}(m, K[\overline{T}]) = msig$ then $\mathsf{smtype}_{\Delta}(m, \sigma K[\overline{T}]) = \sigma\, msig$.*

PROOF. Follows immediately from Lemma 7.1 and the assumption that interfaces of the underlying program are closed. □

Lemma 7.6. *If $\mathsf{fields}(N) = \overline{T\,f}$ then $\mathsf{fields}(\sigma N) = \sigma\overline{T\,f}$.*

PROOF. Straightforward induction on the derivation of $\mathsf{fields}(N) = \overline{T\,f}$. □

Lemma 7.7 (Type substitution preserves expression typing)**.** *Suppose $\Delta \Vdash \sigma\Delta'$ and $\Delta \vdash \sigma X\ \mathsf{ok}$ for all $X \in \mathsf{dom}(\sigma)$ and $\mathsf{dom}(\Delta) \supseteq \mathsf{dom}(\Delta') \setminus \mathsf{dom}(\sigma)$. If $\Delta'; \Gamma \vdash e : T$ then $\Delta; \sigma\Gamma \vdash \sigma e : \sigma T$.*

PROOF. We proceed by induction on the derivation of $\Delta'; \Gamma \vdash e : T$.
Case distinction on the last rule of the derivation of $\Delta'; \Gamma \vdash e : T$.

- *Case* rule EXP-VAR: Obvious.

- *Case* rule EXP-FIELD: Then

$$\frac{\Delta'; \Gamma \vdash e' : C\langle\overline{T}\rangle \qquad \texttt{class } C\langle\overline{X}\rangle \texttt{ extends } N \texttt{ where } \overline{P}\,\{\,\overline{U\,f}\ldots\}}{\Delta'; \Gamma \vdash e'.f_j : [\overline{T/X}]U_j}\ \text{EXP-FIELD}$$

  with $e = e'.f_j$ and $T = [\overline{T/X}]U_j$. Applying the I.H. yields $\Delta; \sigma\Gamma \vdash \sigma e' : C\langle\sigma\overline{T}\rangle$. With rule EXP-FIELD we then get $\Delta; \sigma\Gamma \vdash \sigma(e'.f_j) : [\overline{\sigma T/X}]U_j$. Because the underlying program is well-typed, we have $\mathsf{ftv}(U_j) \subseteq \overline{X}$. Hence, $[\overline{\sigma T/X}]U_j = \sigma[\overline{T/X}]U_j = \sigma T$ as required.

- *Case* rule EXP-INVOKE: Then

$$\frac{\begin{array}{c}\Delta'; \Gamma \vdash e' : T' \qquad \mathsf{mtype}_{\Delta'}(m, T') = \langle\overline{X}\rangle\,\overline{U\,x} \to U \texttt{ where } \overline{\mathcal{P}} \\ (\forall i)\ \Delta'; \Gamma \vdash e_i : [\overline{V/X}]U_i \qquad \Delta' \Vdash [\overline{V/X}]\overline{\mathcal{P}} \qquad \Delta' \vdash \overline{V}\ \mathsf{ok}\end{array}}{\Delta'; \Gamma \vdash e'.m\langle\overline{V}\rangle(\overline{e}) : [\overline{V/X}]U}\ \text{EXP-INVOKE}$$

  with $e = e'.m\langle\overline{V}\rangle(\overline{e})$ and $T = [\overline{V/X}]U$. From the I.H. we get

$$\Delta; \sigma\Gamma \vdash \sigma e' : \sigma T'$$
$$(\forall i)\ \Delta; \sigma\Gamma \vdash \sigma e_i : \sigma[\overline{V/X}]U_i$$

  By Lemma 7.1 we get

$$\Delta \Vdash \sigma[\overline{V/X}]\mathcal{P}$$

  By Lemma 7.3 we get

$$\Delta \vdash \sigma\overline{V}\ \mathsf{ok}$$

  W.l.o.g., $\overline{X}$ fresh, so with Lemma 7.4

$$\mathsf{mtype}_{\Delta}(m, \sigma T') = \langle\overline{X}\rangle\,\overline{\sigma U\,x} \to \sigma U \texttt{ where } \sigma\overline{\mathcal{P}}$$

  With $\overline{X}$ fresh we have $\sigma[\overline{V/X}](\overline{U}, U, \overline{\mathcal{P}}) = [\overline{\sigma V/X}]\sigma(\overline{U}, U, \overline{\mathcal{P}})$, so we may apply rule EXP-INVOKE and get $\Delta; \sigma\Gamma \vdash \sigma e : [\overline{\sigma V/X}]\sigma U$. But $[\overline{\sigma V/X}]\sigma U = \sigma T$ as required.

- *Case* rule EXP-INVOKE-S: Then

$$\frac{\begin{array}{c}\mathsf{smtype}_{\Delta'}(m, I\langle\overline{W}\rangle[\overline{T}]) = \langle\overline{X}\rangle\,\overline{U\,x} \to U \texttt{ where } \overline{\mathcal{P}} \qquad (\forall i)\ \Delta'; \Gamma \vdash e_i : [\overline{V/X}]U_i \\ \Delta' \Vdash [\overline{V/X}]\overline{\mathcal{P}} \qquad 1 \notin \mathsf{pos}^+(I) \text{ or } (\exists i)\ \Delta' \Vdash T_i\,\texttt{mono} \qquad \Delta' \vdash \overline{T}, \overline{V}\ \mathsf{ok}\end{array}}{\Delta'; \Gamma \vdash I\langle\overline{W}\rangle[\overline{T}].m\langle\overline{V}\rangle(\overline{e}) : [\overline{V/X}]U}\ \text{EXP-INVOKE-S}$$

  with $e = I\langle\overline{W}\rangle[\overline{T}].m\langle\overline{V}\rangle(\overline{e})$ and $T = [\overline{V/X}]U$. W.l.o.g., $\overline{X}$ fresh. Hence, by Lemma 7.5

$$\mathsf{smtype}_{\Delta}(m, \sigma I\langle\overline{W}\rangle[\overline{T}]) = \langle\overline{X}\rangle\,\overline{\sigma U\,x} \to \sigma U \texttt{ where } \sigma\overline{\mathcal{P}}$$

  Moreover, $\sigma[\overline{V/X}](\overline{U}, U, \overline{\mathcal{P}}) = [\overline{\sigma V/X}]\sigma(\overline{U}, U, \overline{\mathcal{P}})$. Applying the I.H. then yields

$$(\forall i)\ \Delta; \sigma\Gamma \vdash \sigma e_i : [\overline{\sigma V/X}]\sigma U_i$$

  With Lemma 7.1 we also have

$$\Delta \Vdash [\overline{\sigma V/X}]\sigma\mathcal{P}$$

Moreover, with Lemma 7.3

$$\Delta \vdash \sigma(\overline{T}, \overline{V}) \; \mathsf{ok}$$

Also, if $\Delta' \Vdash T_i \, \mathtt{mono}$ for some i, then with Lemma 7.1

$$\Delta \Vdash \sigma T_i \, \mathtt{mono}$$

We now get with rule EXP-INVOKE-S that $\Delta; \sigma\Gamma \vdash \sigma e : [\overline{\sigma V/X}]\sigma U$. Noting that $[\overline{\sigma V/X}]\sigma U = \sigma T$ finishes this case.

- *Case* rule EXP-NEW: Follows from the I.H., Lemma 7.3, and Lemma 7.6.
- *Case* rule EXP-CAST: Follows from the I.H. and Lemma 7.3.
- *Case* rule EXP-SUBSUME: Follows from the I.H. and Lemma 7.1.

End case distinction on the last rule of the derivation of $\Delta'; \Gamma \vdash e : T$. □

Lemma 7.8 (Expression substitution preserves expression typing)**.** *If $\Delta; \Gamma, x : T \vdash e : U$ and $\Delta; \Gamma : e' : T$ then $\Delta; \Gamma \vdash [e'/x]e : U$.*

PROOF. By induction on the derivation given.
Case distinction on the last rule used in the derivation of $\Delta; \Gamma, x : T \vdash e : U$.

- *Case* rule EXP-VAR: If $e = x$ then $T = U$ and $[e'/x]e = e'$, so the claim follows from the assumptions. Otherwise, $e = y$ for some $y \neq x$ with $(\Gamma, x : T)(y) = U$. Hence, $\Gamma(y) = U$, so the claim follows with rule EXP-VAR.
- *Case* rule EXP-FIELD: Follows from the I.H. and rule EXP-FIELD.
- *Case* rule EXP-INVOKE: Follows from the I.H. and rule EXP-INVOKE.
- *Case* rule EXP-INVOKE-S: Follows from the I.H. and rule EXP-INVOKE-S.
- *Case* rule EXP-NEW: Follows from the I.H. and rule EXP-NEW.
- *Case* rule EXP-CAST: Follows from the I.H. and rule EXP-CAST.
- *Case* rule EXP-SUBSUME: Follows from the I.H. and rule EXP-SUBSUME.

End case distinction on the last rule used in the derivation of $\Delta; \Gamma, x : T \vdash e : U$. □

Lemma 7.9. *If $\emptyset \Vdash \overline{T} \; \mathtt{implements} \; I\langle\overline{V}\rangle$ then one of the following holds:*

- *There exists an implementation definition*

$$\mathtt{implementation}\langle\overline{X}\rangle \; I\langle\overline{V'}\rangle \; [\,\overline{N}\,] \; \mathtt{where} \; \overline{P} \ldots$$

and a substitution $[\overline{U/X}]$ such that $\emptyset \Vdash [\overline{U/X}]\overline{P}$, $\overline{V} = [\overline{U/X}]\overline{V'}$, and $(\forall i)\; \emptyset \vdash T_i \leq [\overline{U/X}]N_i$ with $T_i \neq [\overline{U/X}]N_i$ implying $i \in \mathsf{pos}^-(I)$.

- *$\overline{T} = T$ such that $\emptyset \vdash T \leq J\langle\overline{U}\rangle$, $J\langle\overline{U}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle$, $1 \in \mathsf{pos}^+(J)$, and $1 \in \mathsf{pos}^-(I)$ unless $T = J\langle\overline{U}\rangle$.*

PROOF. From $\emptyset \Vdash \overline{T} \; \mathtt{implements} \; I\langle\overline{V}\rangle$ we get $\emptyset \Vdash_{\mathrm{q}} \overline{T} \; \mathtt{implements} \; I\langle\overline{V}\rangle$ by Theorem 6.36. By Lemma 6.27 we then get the existence of $\overline{T'}$ such that

$$\begin{gathered}\emptyset \Vdash_{\mathrm{q}}' \overline{T'} \; \mathtt{implements} \; I\langle\overline{V}\rangle \\ (\forall i)\; \emptyset \vdash_{\mathrm{q}}' T_i \leq T'_i \\ (\forall i)\; i \in \mathsf{pos}^-(I) \text{ unless } T_i = T'_i \qquad (45)\end{gathered}$$

{eq:pos-minus::lemma:entails-empty-tenv}

By Theorem 6.34 and rule SUB-Q-ALG-KERNEL we have

$$(\forall i)\; \emptyset \vdash T_i \leq T'_i \qquad (46)$$

{eq:Ti-st-Ti'::lemma:entails-empty-tenv}

Case distinction on the last rule of the derivation of $\emptyset \Vdash_{\mathrm{q}}' \overline{T'} \; \mathtt{implements} \; I\langle\overline{V}\rangle$.

- *Case* rule ENT-Q-ALG-ENV: Impossible.
- *Case* rule ENT-Q-ALG-IMPL: Then

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots$$
$$\emptyset \Vdash_{\mathrm{q}} [\overline{U/X}]\overline{P}$$

  with $\overline{V} = [\overline{U/X}]\overline{V'}$ and $\overline{T'} = [\overline{U/X}]\overline{N}$. By Theorem 6.34 we get $\emptyset \Vdash [\overline{U/X}]\overline{P}$. Thus, with (45) and (46), we conclude that the first case of the lemma holds.

- *Case* rule ENT-Q-ALG-IFACE: Then $\overline{T'} = J\langle\overline{U}\rangle$, $1 \in \mathtt{pos}^+(J)$, and $J\langle\overline{U}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle$. With (45) and (46), it is now easy to see that the second case of the lemma holds.

End case distinction on the last rule of the derivation of $\emptyset \Vdash_{\mathrm{q}}{}' \overline{T'}\,\texttt{implements}\,I\langle\overline{V}\rangle$. □

Lemma 7.10. *If $\emptyset \vdash T \leq N$ then either $N = \texttt{Object}$ or $N \neq \texttt{Object}$ and $T = N'$ for some N' with $N' \trianglelefteq_{\mathrm{c}} N$.*

PROOF. If $N = \texttt{Object}$ then we are done. Thus, assume $N \neq \texttt{Object}$. With Theorem 6.34 we have $\emptyset \vdash_{\mathrm{q}} T \leq N$, so $\emptyset \vdash_{\mathrm{q}}{}' T \leq N$ with Lemma 6.16. The claim now follows with Lemma 6.11. □

Lemma 7.11. *If $\emptyset \vdash N \leq I\langle\overline{V}\rangle$ then $N \trianglelefteq_{\mathrm{c}} M$ for some M and there exists a definition $\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [M']\ \texttt{where}\ \overline{P}\ \ldots$ and a substitution $[\overline{U/X}]$ such that $\emptyset \Vdash [\overline{U/X}]\overline{P}$, $\overline{V} = [\overline{U/X}]\overline{V'}$, and $M = [\overline{U/X}]M'$.*

PROOF. From $\emptyset \vdash N \leq I\langle\overline{V}\rangle$ we get $\emptyset \vdash_{\mathrm{q}} N \leq I\langle\overline{V}\rangle$ by Theorem 6.34.
Case distinction on the last rule of the derivation of $\emptyset \vdash_{\mathrm{q}} N \leq I\langle\overline{V}\rangle$.

- *Case* rule SUB-Q-ALG-KERNEL: Then $\emptyset \vdash_{\mathrm{q}}{}' N \leq I\langle\overline{V}\rangle$, which is a contradiction to Lemma 6.11.
- *Case* rule SUB-Q-ALG-IMPL: Hence

$$\emptyset \vdash_{\mathrm{q}}{}' N \leq T$$
$$\emptyset \Vdash_{\mathrm{q}}{}' T\,\texttt{implements}\,I\langle\overline{V}\rangle$$

  By Lemma 6.11 we have $T = M$ for some M with $N \trianglelefteq_{\mathrm{c}} M$. Moreover, the derivation of $\emptyset \Vdash_{\mathrm{q}}{}' M\,\texttt{implements}\,I\langle\overline{V}\rangle$ must end with rule ENT-Q-ALG-IMPL. Inverting this rule and using Theorem 6.36 finishes this case.

End case distinction on the last rule of the derivation of $\emptyset \vdash_{\mathrm{q}} N \leq I\langle\overline{V}\rangle$. □

Lemma 7.12. *If $\emptyset \Vdash T\,\texttt{mono}$ then $T = N$ for some N.*

PROOF. Obvious. □

Lemma 7.13. *If $\Delta; \Gamma \vdash e : T$ then $\mathcal{D} :: \Delta; \Gamma \vdash e : T'$ with $\Delta \vdash T' \leq T$ such that $\mathcal{D}$ does not end with an application of rule* EXP-SUBSUME.

PROOF. Straightforward induction on the derivation of $\Delta; \Gamma \vdash e : T$. □

Lemma 7.14. *If $C\langle\overline{T}\rangle \trianglelefteq_{\mathrm{c}} D\langle\overline{U}\rangle$ then, for fresh and pairwise distinct type variables $\overline{X}$, $C\langle\overline{X}\rangle \trianglelefteq_{\mathrm{c}} D\langle\overline{U'}\rangle$ with $[\overline{T/X}]D\langle\overline{U'}\rangle = D\langle\overline{U}\rangle$.*

PROOF. By induction on the derivation of $C\langle\overline{T}\rangle \trianglelefteq_{\mathrm{c}} D\langle\overline{U}\rangle$.
Case distinction on the last rule in the derivation of $C\langle\overline{T}\rangle \trianglelefteq_{\mathrm{c}} D\langle\overline{U}\rangle$.

- *Case* EXT-C-REFL: Obvious with $\overline{U'} = \overline{X}$.

- *Case* EXT-C-SUPER: Then

$$\frac{\texttt{class}\ C\langle\overline{Y}\rangle\ \texttt{extends}\ C'\langle\overline{V}\rangle\ldots \qquad [\overline{T/Y}]C'\langle\overline{V}\rangle \trianglelefteq_c D\langle\overline{U}\rangle}{C\langle\overline{T}\rangle \trianglelefteq_c D\langle\overline{U}\rangle}$$

By the I.H. there exists $\overline{Z}, \overline{U''}$ with

$$C'\langle\overline{Z}\rangle \trianglelefteq_c D\langle\overline{U''}\rangle$$
$$[\overline{[\overline{T/Y}]V/Z}]D\langle\overline{U''}\rangle = D\langle\overline{U}\rangle$$

We also have for $\sigma = [\overline{X/Y}]$ that $C\langle\overline{X}\rangle \trianglelefteq_c \sigma C'\langle\overline{V}\rangle$. From $C'\langle\overline{Z}\rangle \trianglelefteq_c D\langle\overline{U''}\rangle$ we get with Lemma 6.13 that $[\overline{\sigma V/Z}]C'\langle\overline{Z}\rangle \trianglelefteq_c [\overline{\sigma V/Z}]D\langle\overline{U''}\rangle$. With $[\overline{\sigma V/Z}]C'\langle\overline{Z}\rangle = \sigma C'\langle\overline{V}\rangle$ and Lemma 6.2 we then have

$$C\langle\overline{X}\rangle \trianglelefteq_c [\overline{\sigma V/Z}]D\langle\overline{U''}\rangle$$

Moreover,

$$[\overline{T/X}][\overline{\sigma V/Z}]D\langle\overline{U''}\rangle \overset{\overline{X}\ \text{fresh}}{=} [\overline{[\overline{T/Y}]V/Z}]D\langle\overline{U''}\rangle = D\langle\overline{U}\rangle$$

Define $\overline{U'} = [\overline{\sigma V/Z}]\overline{U''}$ to finish the proof.

End case distinction on the last rule in the derivation of $C\langle\overline{T}\rangle \trianglelefteq_c D\langle\overline{U}\rangle$. □

Lemma 7.15.

(*i*) *If* $\Delta \vdash T\ \mathsf{ok}$ *then* $\mathsf{ftv}(T) \subseteq \mathsf{dom}(\Delta)$.

(*ii*) *If* $\Delta \vdash \mathcal{P}\ \mathsf{ok}$ *then* $\mathsf{ftv}(\mathcal{P}) \subseteq \mathsf{dom}(\Delta)$.

PROOF. We prove the first claim by induction on the derivation of $\Delta \vdash T\ \mathsf{ok}$. The second claim follows from the first one by inverting the last rule in the derivation of $\Delta \vdash \mathcal{P}\ \mathsf{ok}$. □

Lemma 7.16 (Class inheritance propagates well-formedness)**.** *If* $N \trianglelefteq_c M$ *and* $\Delta \vdash N\ \mathsf{ok}$ *then* $\Delta \vdash M\ \mathsf{ok}$.

PROOF. We proceed by induction on the derivation of $N \trianglelefteq_c M$.
Case distinction on the last rule of the derivation of $N \trianglelefteq_c M$.

- *Case* rule EXT-C-REFL: Obvious.

- *Case* rule EXT-C-SUPER: Then

$$\frac{\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N'\ \texttt{where}\ \overline{P}\ldots \qquad [\overline{V/X}]N' \trianglelefteq_c M}{\Delta \vdash C\langle\overline{V}\rangle \leq M}$$

with $N = C\langle\overline{V}\rangle$. Because $\Delta \vdash N\ \mathsf{ok}$, we have $\Delta \Vdash [\overline{V/X}]\overline{P}$ and $\Delta \vdash \overline{V}\ \mathsf{ok}$. The underlying program is well-typed, so $\overline{P}, \overline{X} \vdash N'\ \mathsf{ok}$. With Lemma 7.3 then $\Delta \vdash [\overline{V/X}]N'\ \mathsf{ok}$. Applying the I.H. now yields $\Delta \vdash M\ \mathsf{ok}$.

End case distinction on the last rule of the derivation of $N \trianglelefteq_c M$. □

Lemma 7.17. *If* $\Delta; \Gamma, x : T \vdash e : U$ *and* $\Delta \vdash T' \leq T$ *then* $\Delta; \Gamma, x : T' \vdash e : U$.

PROOF. Straightforward induction on the derivation of $\Delta; \Gamma, x : T \vdash e : U$. □

Lemma 7.18. *If* $N \trianglelefteq_c M$ *then* $\Delta \vdash N \leq M$.

PROOF. The claim is obvious if $M = \texttt{Object}$. Otherwise, it follows using rule SUB-Q-ALG-CLASS, rule SUB-Q-ALG-KERNEL, and Theorem 6.36. $\square$

Lemma 7.19. *Suppose* $\mathsf{mtype}_{\emptyset}(m^{\mathsf{c}}, N) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}$ *and* $\mathsf{getmdef}^{\mathsf{c}}(m^{\mathsf{c}}, N') = \langle\overline{X'}\rangle\,\overline{U'\,x'} \to U'\ \texttt{where}\ \overline{\mathcal{P}'}\,\{e\}$ *and* $\emptyset \vdash N'\ \mathsf{ok}$ *and* $N' \trianglelefteq_{\mathsf{c}} N$ *and* $\emptyset \Vdash \sigma\overline{\mathcal{P}}$ *for some substitution* σ *with* $\mathsf{dom}(\sigma) = \overline{X}$ *and* $\emptyset \vdash \sigma X\ \mathsf{ok}$ *for all* $X \in \mathsf{dom}(\sigma)$.

Then $\overline{X} = \overline{X'}$, $\overline{x} = \overline{x'}$, *and* $\emptyset; \texttt{this} : N', \overline{x : \sigma U} \vdash \sigma e : \sigma U$.

PROOF. In the following, we write simply m instead of m^{c}. The proof is by induction on the derivation of $\mathsf{getmdef}^{\mathsf{c}}(m, N') = \langle\overline{X'}\rangle\,\overline{U'\,x'} \to U'\ \texttt{where}\ \overline{\mathcal{P}'}\,\{e\}$
Case distinction on the last rule used in this derivation.

- *Case* rule DYN-MDEF-C-BASE: Then

$$\frac{\texttt{class}\ C\langle\overline{Z}\rangle\ \texttt{extends}\ M\ \texttt{where}\ \overline{Q}\,\{\ldots\,\overline{m : \mathit{mdef}}\,\} \qquad m = m_k}{\mathsf{getmdef}^{\mathsf{c}}(m, \underbrace{C\langle\overline{T}\rangle}_{=N'}) = \underbrace{[\overline{T/Z}]\mathit{mdef}_k}_{=\langle\overline{X'}\rangle\,\overline{U'\,x'}\to U'\ \texttt{where}\ \overline{\mathcal{P}'}\,\{e\}}}\ \text{DYN-MDEF-C-BASE} \tag{47}$$
{eq:rule::lemma:soundness-alg-subtyping-entailment}

Assume

$$\mathit{mdef}_k = \underbrace{\langle\overline{X'}\rangle\,\overline{U''\,x'} \to U''\ \texttt{where}\ \overline{P''}}_{=\mathit{msig}}\,\{e'\} \tag{48}$$
{eq:mdef-k-def::lemma:soundness-alg-subtyping-entailment}

The underlying program is well-typed, so we have

$$\overline{Q}, \overline{Z} \vdash m_k : \mathit{mdef}_k\ \mathsf{ok\,in}\ C\langle\overline{Z}\rangle$$

Hence,

$$\underbrace{\overline{Q}, \overline{P''}, \overline{Z}, \overline{X'}}_{=\Delta}; \underbrace{\texttt{this} : C\langle\overline{X}\rangle, \overline{x' : U''}}_{=\Gamma} \vdash e' : U'' \tag{49}$$
{eq:type-e'::lemma:soundness-alg-subtyping-entailment}

$$\mathsf{override\text{-}ok}_{\overline{Q},\overline{Z}}(m_k : \mathit{msig}, C\langle\overline{Z}\rangle) \tag{50}$$
{eq:over-ok::lemma:soundness-alg-subtyping-entailment}

Assume $N = D\langle\overline{V}\rangle$. From $C\langle\overline{T}\rangle \trianglelefteq_{\mathsf{c}} D\langle\overline{V}\rangle$ we get with Lemma 7.14 that

$$C\langle\overline{Z}\rangle \trianglelefteq_{\mathsf{c}} D\langle\overline{W}\rangle$$

$$[\overline{T/Z}]D\langle\overline{W}\rangle = D\langle\overline{V}\rangle \tag{51}$$
{eq:w-eq-v::lemma:soundness-alg-subtyping-entailment}

for some $\overline{W}$.
From $\mathsf{mtype}_{\emptyset}(m, D\langle\overline{V}\rangle) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}$ we get

$$\texttt{class}\ D\langle\overline{Z'}\rangle\ \ldots\ \{\ldots\,\overline{m' : \mathit{msig}\{e''\}}\}$$

$$m = m'_j$$

$$\mathit{msig}_j = \langle\overline{X}\rangle\,\overline{U'''\,x} \to U'''\ \texttt{where}\ \overline{P'''} \tag{52}$$
{eq:msig-j-def::lemma:soundness-alg-subtyping-entailment}

$$\langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}} = [\overline{V/Z'}]\mathit{msig}_j \tag{53}$$
{eq:msig-j-eq::lemma:soundness-alg-subtyping-entailment}

Hence, with criterion WF-CLASS-2

$$\mathsf{mtype}_{\overline{Q},\overline{Z}}(m, D\langle\overline{W}\rangle) = [\overline{W/Z'}]\mathit{msig}_j$$

From (47), (50), and rule OK-OVERRIDE

$$\overline{Q}, \overline{Z} \vdash \mathit{msig} \le [\overline{W/Z'}]\mathit{msig}_j \tag{54}$$
{eq:msig-sub::lemma:soundness-alg-subtyping-entailment}

Define

$$\sigma_1 = [\overline{T/Z}]$$
$$\sigma_2 = [\overline{V/Z'}]$$
$$\sigma_3 = [\overline{W/Z'}]$$

We then have from (48), (52) and (54) that

$$\overline{X} = \overline{X'}$$
$$\overline{x} = \overline{x'}$$
$$\overline{U''} = \sigma_3 \overline{U'''} \quad (55) \quad \{\texttt{eq:u'''-eq::lemma:soundness-alg-subtyping-entailment}\}$$
$$\overline{P''} = \sigma_3 \overline{P'''} \quad (56) \quad \{\texttt{eq:p''-eq::lemma:soundness-alg-subtyping-entailment}\}$$
$$\Delta \vdash U'' \leq \sigma_3 U''' \quad (57) \quad \{\texttt{eq:u''-sub::lemma:soundness-alg-subtyping-entailment}\}$$

From the assumption $\emptyset \vdash C\langle\overline{T}\rangle$ ok we get that $\emptyset \Vdash \sigma_1\overline{Q}$ (by inverting rule OK-CLASS) and that $\mathsf{ftv}(\overline{T}) = \emptyset$. (by Lemma 7.15). The underlying program is well-typed, so $\mathsf{ftv}(\overline{Q}) \subseteq \overline{Z}$, so $\sigma\sigma_1\overline{Q} = \sigma_1\overline{Q}$ by definition of σ_1. Hence

$$\emptyset \Vdash \sigma\sigma_1\overline{Q} \quad (58) \quad \{\texttt{eq:entails-q::lemma:soundness-alg-subtyping-entailment}\}$$

We have $\emptyset \Vdash \sigma\overline{\mathcal{P}}$ by assumption. Moreover,

$$\sigma\overline{\mathcal{P}} \overset{(52),(53)}{=} \sigma\sigma_2\overline{P'''} \overset{(51)}{=} \sigma[\overline{\sigma_1 W/Z'}]\overline{P'''} \overset{\text{w.l.o.g.},\overline{Z}\cap\mathsf{ftv}(\overline{P'''})=\emptyset}{=} \sigma\sigma_1\sigma_3\overline{P'''} \overset{(56)}{=} \sigma\sigma_1\overline{P''}$$

Hence,

$$\emptyset \Vdash \sigma\sigma_1\overline{P''} \quad (59) \quad \{\texttt{eq:entails-p''::lemma:soundness-alg-subtyping-entailment}\}$$

Noting that $\mathsf{ftv}(\overline{T}) = \emptyset$, we see that $\sigma\sigma_1 = [\overline{\sigma X/X}, \overline{T/Z}]$. Thus, with $\overline{X} = \overline{X'}$

$$\mathsf{dom}(\Delta) \setminus \mathsf{dom}(\sigma\sigma_1) = \emptyset$$

Moreover, from $\emptyset \vdash C\langle\overline{T}\rangle$ ok we have $\emptyset \vdash \overline{T}$ ok, so with the assumptions we get

$$\emptyset \vdash \sigma\sigma_1 Y \ \mathsf{ok} \text{ for all } Y \in \mathsf{dom}(\sigma\sigma_1)$$

Hence, we may apply Lemma 7.7 to (49) and get

$$\emptyset; \sigma\sigma_1\Gamma \vdash \sigma\sigma_1 e' : \sigma\sigma_1 U'' \quad (60) \quad \{\texttt{eq:1::lemma:soundness-alg-subtyping-entailment}\}$$

With $\mathsf{ftv}(\overline{T}) = \emptyset$, we have $\sigma\sigma_1 N' = N'$. Moreover,

$$\sigma\sigma_1 U''_i \overset{(55)}{=} \sigma\sigma_1\sigma_3 U'''_i \overset{\text{w.l.o.g.},\overline{Z}\cap\mathsf{ftv}(\overline{U'''})=\emptyset}{=} \sigma[\overline{\sigma_1 W/Z'}]U'''_i \overset{(51)}{=} \sigma\sigma_2 U'''_i \overset{(53)}{=} \sigma U_i$$

Hence,

$$\sigma\sigma_1\Gamma = \texttt{this} : N', \overline{x : \sigma U} \quad (61) \quad \{\texttt{eq:2::lemma:soundness-alg-subtyping-entailment}\}$$

We also have from (47) and (48)

$$\sigma\sigma_1 e' = \sigma e \quad (62) \quad \{\texttt{eq:3::lemma:soundness-alg-subtyping-entailment}\}$$

With (57), (58), (59) and Lemma 7.1 we get

$$\emptyset \vdash \sigma\sigma_1 U'' \leq \sigma\sigma_1\sigma_3 U'''$$

We also have

$$\sigma\sigma_1\sigma_3 U''' \overset{\text{w.l.o.g.},\overline{Z}\cap\mathsf{ftv}(U''')=\emptyset}{=} \sigma[\overline{\sigma_1 W/Z'}]U'' \overset{(51)}{=} \sigma\sigma_2 U''' \overset{(53)}{=} \sigma U$$

Hence,

$$\emptyset \vdash \sigma\sigma_1 U'' \leq \sigma U$$

With (60), (61), (62), and rule EXP-SUBSUME then

$$\emptyset; \mathtt{this} : N', \overline{x : \sigma U} \vdash \sigma e : \sigma U$$

as required.

- *Case* rule DYN-MDEF-C-SUPER: Then

$$\frac{\begin{array}{c}\mathtt{class}\ C\langle\overline{Z}\rangle\ \mathtt{extends}\ M\ \mathtt{where}\ \overline{Q}\,\{\ldots\ \overline{m : mdef}\,\} \\ m \notin \overline{m} \qquad \mathsf{getmdef}^{\mathrm{c}}(m, [\overline{T/Z}]M) = \langle\overline{X'}\rangle\,\overline{U'\,x'} \to U'\ \mathtt{where}\ \overline{\mathcal{P}'}\,\{e\}\end{array}}{\mathsf{getmdef}^{\mathrm{c}}(m, C\langle\overline{T}\rangle) = \langle\overline{X'}\rangle\,\overline{U'\,x'} \to U'\ \mathtt{where}\ \overline{\mathcal{P}'}\,\{e\}}\ \text{DYN-MDEF-C-SUPER}$$

  with $N' = C\langle\overline{T}\rangle$.

  Assume $[\overline{T/Z}]M \not\trianglelefteq_{\mathrm{c}} N$. Then, because $N' \trianglelefteq_{\mathrm{c}} N$, we must have $N' = N$. But with $\mathsf{mtype}_{\emptyset}(m^{\mathrm{c}}, N) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \mathtt{where}\ \overline{\mathcal{P}}$ we then have $m \in \overline{m}$, which is a contradiction.

  Thus, $[\overline{T/Z}]M \trianglelefteq_{\mathrm{c}} N$. Obviously, also $N' \trianglelefteq_{\mathrm{c}} [\overline{T/Z}]M$, so with Lemma 7.16 $\emptyset \vdash [\overline{T/Z}]M\ \mathsf{ok}$. Thus, we can apply the I.H. and get

$$\overline{X} = \overline{X'}$$
$$\overline{x} = \overline{x'}$$
$$\emptyset; \mathtt{this} : [\overline{T/Z}]M, \overline{x : \sigma U} \vdash \sigma e : \sigma U$$

  By Lemma 7.18 $\emptyset \vdash N' \leq [\overline{T/Z}]M$, so an application of Lemma 7.17 finishes this case.

End case distinction on the last rule used in this derivation. □

Lemma 7.20. *If* $\mathsf{mtype}_{\emptyset}(m^{\mathrm{c}}, N) = \langle\overline{X}^n\rangle\,\overline{U\,x}^m \to U\ \mathtt{where}\ \overline{\mathcal{P}}$ *and* $N' \trianglelefteq_{\mathrm{c}} N$ *then* $\mathsf{getmdef}^{\mathrm{c}}(m^{\mathrm{c}}, N') = \langle\overline{Y}^n\rangle\,\overline{V\,y}^m \to V\ \mathtt{where}\ \overline{\mathcal{Q}}\,\{e\}$.

PROOF. We proceed by induction on the derivation of $N' \trianglelefteq_{\mathrm{c}} N$.
Case distinction on the last rule used in the derivation of $N' \trianglelefteq_{\mathrm{c}} N$.

- *Case* rule EXT-C-REFL: Then $N' = N$ and the claim follows with rule DYN-MDEF-C-BASE and criterion WF-CLASS-2.

- *Case* rule EXT-C-SUPER: Then

$$\frac{\mathtt{class}\ C\langle\overline{X}\rangle\ \mathtt{extends}\ M\ \mathtt{where}\ \overline{P'}\,\{\ldots\ \overline{m : mdef}\,\} \qquad [\overline{T/X}]M \trianglelefteq_{\mathrm{c}} N}{C\langle\overline{T}\rangle \trianglelefteq_{\mathrm{c}} N}\ \text{EXT-C-SUPER}$$

  with $N' = C\langle\overline{T}\rangle$.

  - Assume $m^{\mathrm{c}} \notin \overline{m}$. We get by the I.H. that $\mathsf{getmdef}^{\mathrm{c}}(m^{\mathrm{c}}, [\overline{T/X}]M) = \langle\overline{Y}^n\rangle\,\overline{V\,y}^m \to V\ \mathtt{where}\ \overline{\mathcal{Q}}\,\{e\}$. With $m^{\mathrm{c}} \notin \overline{m}$ we then have $\mathsf{getmdef}^{\mathrm{c}}(m^{\mathrm{c}}, C\langle\overline{T}\rangle) = \langle\overline{Y}^n\rangle\,\overline{V\,y}^m \to V\ \mathtt{where}\ \overline{\mathcal{Q}}\,\{e\}$ by rule DYN-MDEF-C-SUPER.
  - Assume $m^{\mathrm{c}} \in \overline{m}$. Then $\mathsf{getmdef}^{\mathrm{c}}(m^{\mathrm{c}}, C\langle\overline{T}\rangle) = \langle\overline{Y}^{n'}\rangle\,\overline{V\,y}^{m'} \to V\ \mathtt{where}\ \overline{\mathcal{Q}}\,\{e\}$ and, by rule MTYPE-CLASS, $\mathsf{mtype}_{\Delta}(m^{\mathrm{c}}, C\langle\overline{T}\rangle) = \langle\overline{Y}^{n'}\rangle\,\overline{V\,y}^{m'} \to V\ \mathtt{where}\ \overline{\mathcal{Q}}\,\{e\}$. Because the underlying program is well-typed, we know that method m^{c} of class C correctly overrides method m^{c} of class D, where $N = D\langle\overline{W}\rangle$. But this implies $n = n'$ and $m = m'$ as required.

End case distinction on the last rule used in the derivation of $N' \trianglelefteq_{c} N$. $\square$

Lemma 7.21. *If* $N \trianglelefteq_{c} C\langle\overline{T}\rangle$ *and* `class` $C\langle\overline{X}\rangle \ldots \{\overline{U\,f}\ \ldots\}$ *and* $\mathsf{fields}(N) = \overline{V\,g}$*, then* $\overline{V\,g} = \overline{V'\,g'}\ ([\overline{T/X}]\overline{U\,f})\ \overline{V''\,g''}$

Proof. Straightforward induction on the derivation of $N \trianglelefteq_{c} C\langle\overline{T}\rangle$. $\square$

Lemma 7.22. *If* $\mathsf{fields}(N) = \overline{U\,f}^{n}$ *and* $i, j \in [n]$ *with* $i \neq j$*, then* $f_i \neq f_j$*.*

Proof. Follows by induction on the derivation of $\mathsf{fields}(N) = \overline{U\,f}$, using criterion Wf-Class-1. $\square$

Lemma 7.23. *For all* N*, there exist* $\overline{U}$ *and* $\overline{f}$ *such that* $\mathsf{fields}(N) = \overline{U\,f}$*.*

Proof. Assume $N = C\langle\overline{T}\rangle$. The claim now follows by induction on the depth of C in the inheritance tree. $\square$

Lemma 7.24. *If* $\emptyset \Vdash \overline{T}\ \texttt{implements}\ I\langle\overline{V}\rangle$ *and there exists* j *with* $\emptyset \vdash M \leq T_j$ *for some* M*, then there exists a definition*

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P} \ldots$$

and a substitution $[\overline{U/X}]$ *such that*

(i) $\emptyset \Vdash [\overline{U/X}]\overline{P}$*;*

(ii) $\overline{V} = [\overline{U/X}]\overline{V'}$*;*

(iii) $\emptyset \vdash M \leq [\overline{U/X}]N_j$

(iv) *if* $j \notin \mathsf{pos}^{+}(I)$ *then* $\emptyset \vdash T_j \leq [\overline{U/X}]N_j$ *with* $T_j \neq [\overline{U/X}]N_j$ *implying* $j \in \mathsf{pos}^{-}(I)$*;*

(v) *if* $j \in \mathsf{pos}^{+}(I)$ *and* $j \notin \mathsf{pos}^{-}(I)$ *and* $T_j \neq [\overline{U/X}]N_j$*, then* $\overline{T} = T_j = J\langle\overline{W}\rangle$ *with* $J\langle\overline{W}\rangle \trianglelefteq_{i} I\langle\overline{V}\rangle$ *and* $1 \in \mathsf{pos}^{+}(J)$*;*

(vi) $(\forall i \neq j)\ \emptyset \vdash T_i \leq [\overline{U/X}]N_i$ *with* $T_i \neq [\overline{U/X}]N_i$ *implying* $i \in \mathsf{pos}^{-}(I)$*.*

Proof. By Lemma 7.9, there are two possibilities. The first of these possibilities implies the existence of a definition

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P} \ldots$$

and a substitution $[\overline{U/X}]$ such that

- $\emptyset \Vdash [\overline{U/X}]\overline{P}$
- $\overline{V} = [\overline{U/X}]\overline{V'}$
- $(\forall i)\ \emptyset \vdash T_i \leq [\overline{U/X}]N_i$ with $T_i \neq [\overline{U/X}]N_i$ implying $i \in \mathsf{pos}^{-}(I)$.

With $\emptyset \vdash M \leq T_j$ we then also have $\emptyset \vdash M \leq [\overline{U/X}]N_j$ by transitivity of subtyping. Claim (v) also holds because it is impossible to have $j \notin \mathsf{pos}^{-}(I)$ and $T_j \neq [\overline{U/X}]N_j$ at the same time.

Now assume that the second possibility holds. That is,

$$\begin{gathered}\overline{T} = T \\ \emptyset \vdash T \leq J\langle\overline{W}\rangle \\ J\langle\overline{W}\rangle \trianglelefteq_{i} I\langle\overline{V}\rangle \\ 1 \in \mathsf{pos}^{+}(J) \\ 1 \in \mathsf{pos}^{-}(I) \text{ unless } T = J\langle\overline{W}\rangle\end{gathered}$$

This implies $j = 1$. By transitivity of subtyping, we have $\emptyset \vdash M \leq I\langle\overline{V}\rangle$ Hence, with Lemma 7.11, we know that there exists M' such that

$$\begin{gathered}M \trianglelefteq_c M' \\ \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\,N\,]\ \texttt{where}\ \overline{P}\ldots \\ \emptyset \Vdash [\overline{U/X}]\overline{P} \\ \overline{V} = [\overline{U/X}]\overline{V'} \\ M' = [\overline{U/X}]N\end{gathered}$$

We then have $\emptyset \vdash M \leq [\overline{U/X}]N$ by Lemma 7.18, so claim (iii) holds. Moreover, we get from $1 \in \mathsf{pos}^+(J)$ and Lemma 6.20 that $1 \in \mathsf{pos}^+(I)$, so claim (iv) holds. Now assume $1 \notin \mathsf{pos}^-(I)$. Then $T = J\langle\overline{W}\rangle$, so claim (v) holds. Claim (vi) holds trivially. Setting $\overline{N} = N$ finishes the proof. □

Lemma 7.25. *If $\emptyset \Vdash \overline{T}\,\texttt{implements}\,I\langle\overline{V}\rangle$ and there exists j with $j \notin \mathsf{pos}^+(I)$, then there exists a definition*

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ldots$$

such that

- $\emptyset \Vdash [\overline{U/X}]\overline{P}$
- $\overline{V} = [\overline{U/X}]\overline{V'}$
- $(\forall i)\ \emptyset \vdash T_i \leq [\overline{U/X}]N_i$ *with* $T_i \neq [\overline{U/X}]N_i$ *implying* $i \in \mathsf{pos}^-(I)$

Proof. By Lemma 7.9, there are two possibilities. The first of these possibilities directly implies the claim.

Now assume that the second possibility holds. That is, $\overline{T} = T$ and $1 \in \mathsf{pos}^+(J)$. Hence, $j = 1$. With Lemma 6.20 we then get $1 \in \mathsf{pos}^+(I)$. But this contradicts the assumption $1 \notin \mathsf{pos}^+(I)$. □

Lemma 7.26. *If $N \trianglelefteq_c N_1$ and $N \trianglelefteq_c N_2$ then either $N_1 \trianglelefteq_c N_2$ or $N_2 \trianglelefteq_c N_1$.*

Proof. By straightforward induction on the derivations given. □

Lemma 7.27. *Let*

$$\begin{aligned}\mathscr{M} = \{&(\sigma, \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}^l\,]\ \ldots) \\ &\mid \mathsf{dom}(\sigma) = \overline{X}, (\forall i \in [l])\ M_i^? = \mathsf{nil}\ \textit{or}\ M_i^? \trianglelefteq_c \sigma N_i\}\end{aligned}$$

If $\mathscr{M} \neq \emptyset$, $\mathscr{M}$ finite, and $i \in \mathsf{disp}(I)$ implies $M_i^? \neq \mathsf{nil}$ for all $i \in [l]$, then there exist $(\sigma, impl)$ such that $\mathsf{minimpl}\,\mathscr{M} = (\sigma, impl)$.

Proof. Assume

$$\begin{gathered}\mathscr{M} = \{(\sigma_1, impl_1), \ldots, (\sigma_n, impl_n)\} \\ (\forall i \in [n])\ impl_i = \texttt{implementation}\langle\overline{X_i}\rangle\ I\langle\overline{V_i}\rangle\ [\,\overline{N_i}^l\,]\ \ldots\end{gathered}$$

We then need to show that there exists some $k \in [n]$ such that

$$(\forall i \in [n])\ \sigma_k \overline{N_k}^l \trianglelefteq_c \sigma_i \overline{N_i}^l$$

(The notation $\overline{N}^l \trianglelefteq_c \overline{M}^l$ is short for $(\forall i \in [l])\ N_i \trianglelefteq_c M_i$.)

We proceed by induction on n.

- $n = 1$. Obvious because subtyping is reflexive.

- $n > 1$. Assume

$$\mathscr{M}' = \{(\sigma_1, impl_1), \ldots, (\sigma_{n-1}, impl_{n-1})\}$$

such that that $\mathscr{M} = \mathscr{M}' \cup \{(\sigma_n, impl_n)\}$. By the I.H. we know that there exists $k' \in [n-1]$ such that

$$(\forall i \in [n-1])\ \sigma_{k'} \overline{N_{k'}} \trianglelefteq_c \sigma_i \overline{N_i} \tag{63}$$ {eq:IH::lemma:minimpl-exists}

Now consider $impl_n$. We partition $[l]$ into $[l] = \mathscr{L}_1 \dot\cup \mathscr{L}_2 \dot\cup \mathscr{L}_3$ such that

$$\begin{array}{ll} (\forall j \in \mathscr{L}_1) & \sigma_n N_{nj} \trianglelefteq_c \sigma_{k'} N_{k'j} \\ (\forall j \in \mathscr{L}_2) & \sigma_n N_{nj} \ntrianglelefteq_c \sigma_{k'} N_{k'j} \text{ but } \sigma_{k'} N_{k'j} \trianglelefteq_c \sigma_n N_{nj} \\ (\forall j \in \mathscr{L}_3) & \sigma_n N_{nj} \ntrianglelefteq_c \sigma_{k'} N_{k'j} \text{ and } \sigma_{k'} N_{k'j} \ntrianglelefteq_c \sigma_n N_{nj} \end{array} \tag{64}$$ {eq:def-L123::lemma:minimpl-exists}

We first show that $j \in \mathscr{L}_3$ implies $j \notin \mathsf{disp}(I)$. For the sake of a contradiction, assume $j \in \mathscr{L}_3$ and $j \in \mathsf{disp}(I)$. Then $M_j^? \neq \mathsf{nil}$, so we have

$$\begin{array}{l} M_j^? \trianglelefteq_c \sigma_n N_{nj} \\ M_j^? \trianglelefteq_c \sigma_{k'} N_{k'j} \end{array}$$

By Lemma 7.26 we then have either $\sigma_n N_{nj} \trianglelefteq_c \sigma_{k'} N_{k'j}$ or $\sigma_{k'} N_{k'j} \trianglelefteq_c \sigma_N N_{nj}$. But this is a contradiction to the definition of $\mathscr{L}_3$. Thus, we have shown that

$$j \in \mathscr{L}_3 \text{ implies } j \notin \mathsf{disp}(I) \tag{65}$$ {eq:not-in-disp::lemma:minimpl-exists}

Next, we define for $j \in \mathscr{L}_1 \cup \mathscr{L}_2 \cup \mathscr{L}_3$:

$$M_j = \begin{cases} \sigma_n N_{nj} & \text{if } j \in \mathscr{L}_1 \\ \sigma_{k'} N_{k'j} & \text{if } j \in \mathscr{L}_2 \\ \sigma_n N_{nj} & \text{if } j \in \mathscr{L}_3 \end{cases} \tag{66}$$ {eq:def-Mj::lemma:minimpl-exists}

We then have by definition of $\mathscr{L}_1$ and $\mathscr{L}_2$ that

$$(\forall j \in \mathscr{L}_1 \cup \mathscr{L}_2)\ \emptyset \vdash \sigma_n N_{nj} \sqcap \sigma_{k'} N_{k'j} = M_j$$

Moreover, from (65) we have that $j \in \mathsf{disp}(I)$ implies $j \notin \mathscr{L}_3$ which in turn implies $j \in \mathscr{L}_1 \cup \mathscr{L}_2$. Thus, criterion WF-PROG-2 yields $\sigma_n N_{nj} = \sigma_{k'} N_{k'j}$ for all $j \notin \mathsf{disp}(I)$, so we have with (65) that

$$(\forall j \in \mathscr{L}_3)\ \sigma_n N_{nj} = \sigma_{k'} N_{k'j} \tag{67}$$ {eq:L3::lemma:minimpl-exists}

Thus, we have

$$\emptyset \vdash \sigma_n \overline{N_n}^l \sqcap \sigma_{k'} \overline{N_{k'}}^l = \overline{M}^l$$

By criterion WF-PROG-3 we get the existence of a definition

$$impl = \texttt{implementation}\langle \overline{Y} \rangle\ I \langle \overline{V'} \rangle\ [\,\overline{M'}\,]\ \ldots$$

and a substitution τ with $\mathsf{dom}(\tau) = \overline{Y}$ such that $\tau \overline{M'} = \overline{M}$. By construction of $\overline{M}$, we know that

$$(\tau, impl) \in \mathscr{M} \tag{68}$$ {eq:impl-in-M::lemma:minimpl-exists}

Moreover, we have for all $i \in [n-1]$, $j \in [l] = \mathscr{L}_1 \cup \mathscr{L}_2 \cup \mathscr{L}_3$ that

$$\tau M'_j = M_j \overset{(66)}{=} \begin{cases} \sigma_n N_{nj} \overset{(64)}{\trianglelefteq_c} \sigma_{k'} N_{k'j} \overset{(63)}{\trianglelefteq_c} \sigma_i N_{ij} & \text{if } j \in \mathscr{L}_1 \\ \sigma_{k'} N_{k'j} \overset{(63)}{\trianglelefteq_c} \sigma_i N_{ij} & \text{if } j \in \mathscr{L}_2 \\ \sigma_n N_{nj} \overset{(67)}{=} \sigma_{k'} N_{k'j} \overset{(63)}{\trianglelefteq_c} \sigma_i N_{ij} & \text{if } j \in \mathscr{L}_3 \end{cases}$$

Moreover, we have for all $j \in [l] = \mathscr{L}_1 \cup \mathscr{L}_2 \cup \mathscr{L}_3$

$$\tau M'_j = M_j \overset{(66)}{=} \begin{cases} \sigma_n N_{nj} & \text{if } j \in \mathscr{L}_1 \\ \sigma_{k'} N_{k'j} \overset{(64)}{\trianglelefteq_c} \sigma_n N_{nj} & \text{if } j \in \mathscr{L}_2 \\ \sigma_n N_{nj} & \text{if } j \in \mathscr{L}_3 \end{cases}$$

Thus,

$$(\forall i \in [n], j \in [l])\ \tau M'_j \trianglelefteq_c \sigma_i N_{ij}$$

Finally, with (68) and rule MIN-MDEF, we get

$$\mathsf{minimpl}\mathscr{M} = (\tau, \mathit{impl})$$ □

Lemma 7.28. *Let*

$$\begin{aligned} \mathscr{M} = \{&(\sigma, \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}^l]\ \ldots) \\ &\mid \mathsf{dom}(\sigma) = \overline{X}, (\forall i \in [l])\ N_i = \texttt{Object}\ \textit{or}\ M_i \trianglelefteq_c \sigma N_i\} \end{aligned}$$

If $\mathscr{M} \neq \emptyset$ and $\mathscr{M}$ finite, then there exist (σ, impl) such that $\mathsf{minimpl}\mathscr{M} = (\sigma, \mathit{impl})$.

PROOF. Assume

$$\mathscr{M} = \{(\sigma_1, \mathit{impl}_1), \ldots, (\sigma_n, \mathit{impl}_n)\}$$
$$(\forall i \in [n])\ \mathit{impl}_i = \texttt{implementation}\langle\overline{X_i}\rangle\ I\langle\overline{V_i}\rangle\ [\overline{N_i}^l]\ \ldots$$

Then we have for all $i \in [n]$ and all $j \in [l]$:

$$N_{ij} = \texttt{Object} \text{ or } M_j \trianglelefteq_c \sigma_i N_{ij}$$

Now define

$$\mathscr{L}_1 := \{j \in [l] \mid \text{ there exists } i \in [n], M_j \trianglelefteq_c \sigma_i N_{ij}\}$$
$$\mathscr{L}_2 := [l] \setminus \mathscr{L}_1 = \{j \in [l] \mid \text{ for all } i \in [n], N_{ij} = \texttt{Object}\}$$
$$(\forall j \in [l])\ M'_j = \begin{cases} M_j & \text{if } j \in \mathscr{L}_1 \\ \texttt{Object} & \text{if } j \in \mathscr{L}_2 \end{cases}$$

We now show for

$$\begin{aligned} \mathscr{M}' = \{&(\sigma, \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}^l]\ \ldots) \\ &\mid \mathsf{dom}(\sigma) = \overline{X}, (\forall i \in [l])\ M'_i \trianglelefteq_c \sigma N_i\} \end{aligned}$$

that $\mathscr{M} = \mathscr{M}'$. The claim then follows with Lemma 7.27.

- "$\mathscr{M} \subseteq \mathscr{M}'$". Assume $(\sigma, \mathit{impl}) \in \mathscr{M}$, that is, $(\sigma, \mathit{impl}) = (\sigma_i, \mathit{impl}_i)$ for some $i \in [n]$. Then

  $$(\forall j \in [l])\ M'_j \trianglelefteq_c \sigma_i N_{ij}$$

  by construction of M'_j. Then $(\sigma, \mathit{impl}) \in \mathscr{M}'$.

- "$\mathscr{M} \supseteq \mathscr{M}'$". Assume $(\sigma, impl) \in \mathscr{M}'$ with

$$impl = \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}^{l}\ ]\ \ldots$$

  Then $(\forall i \in [l])\ M_i' \trianglelefteq_c \sigma N_i$. Suppose $j \in [l]$. If $M_j' = \texttt{Object}$ then $N_j = \texttt{Object}$. Otherwise, $M_j' = M_j$, so $M_j \trianglelefteq_c \sigma N_j$. Hence, $(\sigma, impl) \in \mathscr{M}$. □

Lemma 7.29. *If $\Delta; \Gamma \vdash \texttt{new}\, N(\overline{e}) : T$ then $\Delta \vdash N \leq T$ and $\Delta \vdash N$ $\texttt{ok}$.*

Proof. By Lemma 7.13 we have $\mathcal{D} :: \Delta; \Gamma \vdash \texttt{new}\, N(\overline{e}) : T'$ such that $\Delta \vdash T' \leq T$ and $\mathcal{D}$ does not end with rule exp-subsume. Thus, $\mathcal{D}$ ends with rule exp-new. Inverting the rule yields $T' = N$ and $\Delta \vdash N$ $\texttt{ok}$ □

Lemma 7.30. *If $\Delta \vdash N \leq M$ then $N \trianglelefteq_c M$*

Proof. By Theorem 6.34 we have $\Delta \vdash_q N \leq M$, so $\Delta \vdash_q{}' N \leq M$ by Lemma 6.16. The claim now follows with Lemma 6.11. □

Lemma 7.31. *If $M_1 \trianglelefteq_c N$ and $M_2 \trianglelefteq_c N$ then $M_1 \sqcup M_2 \trianglelefteq_c N$.*

Proof. By induction on the derivation of $M_1 \trianglelefteq_c N$.
Case distinction on the last rule of the derivation of $M_1 \trianglelefteq_c N$.

- *Case* rule ext-c-refl: Then $M_1 = N$ and $M_1 \sqcup M_2 = M_2$, so the claim holds.
- *Case* rule ext-c-super: Then

$$\frac{\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ M_1'\ \ldots \qquad [\overline{T/X}]M_1' \trianglelefteq_c N}{C\langle\overline{T}\rangle \trianglelefteq_c N}\ \textsc{ext-c-super}$$

  with $M_1 = C\langle\overline{T}\rangle$. Applying the I.H. yields

$$[\overline{T/X}]M_1' \sqcup M_2 \trianglelefteq_c N$$

  The claim holds obviously if $M_1 \trianglelefteq_c M_2$ or $M_2 \trianglelefteq_c M_1$. Otherwise, we have

$$M_1 \sqcup M_2 = [\overline{T/X}]M_1' \sqcup M_2$$

  by rule lub-super, so the claim also holds.

End case distinction on the last rule of the derivation of $M_1 \trianglelefteq_c N$. □

Lemma 7.32. *If $M_i \trianglelefteq_c N$ for all $i \in [n]$ with $n > 0$, then $\bigsqcup\{M_1, \ldots, M_n\} \trianglelefteq_c N$.*

Proof. We proceed by induction on n.

- $n = 1$. Then $\bigsqcup\{M_1, \ldots, M_n\} = M_1$ and the claim is obvious.
- $n > 1$. By the I.H. we know that

$$\bigsqcup\{M_1, \ldots, M_{n-1}\} \trianglelefteq_c N$$

  By inverting rule lub-set-multi we get

$$\bigsqcup\{M_1, \ldots, M_{n-1}\} \sqcup M_n = \bigsqcup\{M_1, \ldots, M_n\}$$

  The claim now follows from the assumption $M_n \trianglelefteq_c N$ and Lemma 7.31. □

Definition 7.33 (Stuck on a bad cast)**.** *An expression e is* stuck on a bad cast *if and only if there exists an evaluation context $\mathcal{E}$, a class type N, and a value $v = \texttt{new}\, M(\overline{w})$ such that $e = \mathcal{E}[(N)\, v]$ and $M \not\trianglelefteq_c N$.*

Theorem 7.34 (Progress)**.** *If $\emptyset; \emptyset \vdash e : T$ then either $e = v$ for some value v or $e \longrightarrow e'$ for some e' or e is stuck on a bad cast.*

Proof. By induction on the derivation of $\emptyset; \emptyset \vdash e : T$.
Case distinction on the last rule of the derivation of $\emptyset; \emptyset \vdash e : T$.

- *Case* rule EXP-VAR: Impossible.
- *Case* rule EXP-FIELD: Then

$$\frac{\emptyset; \emptyset \vdash e_0 : C\langle\overline{T}\rangle \qquad \texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\,\{\,\overline{U\,f}\ldots\}}{\emptyset; \emptyset \vdash e_0.f_j : [\overline{T/X}]U_j}\ \textsc{exp-field}$$

  with $T = [\overline{T/X}]U_j$. Applying the I.H. to $\emptyset; \emptyset \vdash e_0 : C\langle\overline{T}\rangle$ leaves us with three cases:

  1. $e_0 = v$ for some v. Then $v = \texttt{new}\, D\langle\overline{V}\rangle(\overline{v})$ and $\emptyset \vdash D\langle\overline{V}\rangle \leq C\langle\overline{T}\rangle$ by Lemma 7.13. By Lemma 7.10 then $D\langle\overline{V}\rangle \trianglelefteq_c C\langle\overline{T}\rangle$. By Lemma 7.23, there exists $\overline{W}$ and $\overline{g}$ such that $\mathsf{fields}(D\langle\overline{V}\rangle) = \overline{W\,g}$. By Lemma 7.21 and Lemma 7.22 we know that there exists a unique i such that $W_i\, g_i = [\overline{T/X}]U_j\, f_j$. Hence, $v.f_j \longrightarrow v_i$ by rule DYN-FIELD and rule DYN-CONTEXT.
  2. $e_0 \longrightarrow e_0'$ for some e_0'. It is easy to see that in this case also $e_0.f_j \longrightarrow e_0'.f_j$.
  3. e_0 is stuck on a bad cast. Then $e_0.f_j$ is also stuck on a bad cast.

- *Case* rule EXP-INVOKE: Then

$$\frac{\begin{array}{c}\emptyset; \emptyset \vdash e_0 : T_0 \qquad \mathsf{mtype}_{\emptyset}(m, T_0) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}} \\ (\forall i \in [n])\ \emptyset; \emptyset \vdash e_i : [\overline{V/X}]U_i \qquad \emptyset \Vdash [\overline{V/X}]\overline{\mathcal{P}} \qquad \emptyset \vdash \overline{V}\ \mathsf{ok}\end{array}}{\emptyset; \emptyset \vdash \underbrace{e_0.m\langle\overline{V}\rangle(\overline{e}^n)}_{=e} : \underbrace{[\overline{V/X}]U}_{=T}}\ \textsc{exp-invoke} \tag{69}$$

{eq:invoke-d::theorem:progress}

  We now apply to I.H. to $\emptyset; \emptyset \vdash e_i : T_i$ (for $i = 0, \ldots, n$). This leaves us with three possibilities:

  1. There exist $v_0, \ldots, v_n$ such that $e_i = v_i$ for all $i = 0, \ldots, n$. We deal with this case shortly.
  2. There exist some $m < n$ and some $v_0, \ldots, v_m$ such that $e_i = v_i$ for all $i = 0, \ldots, m$, and $e_{m+1} \longrightarrow e'_{m+1}$. It is easy to see that in this case e also makes an evaluation step.
  3. There exist some $m < n$ and some $v_0, \ldots, v_m$ such that $e_i = v_i$ for all $i = 0, \ldots, m$, and e_{m+1} is stuck on a bad cast. In this case, e is also stuck on a bad cast.

  We now deal with the case that there exist $v_0, \ldots, v_n$ such that $e_i = v_i$ for all $i = 0, \ldots, n$. Assume

$$e_i = v_i = \texttt{new}\, N_i(\overline{w_i}) \quad \text{for } i = 0, \ldots, n \tag{70}$$

{eq:ei-vi::theorem:progress}

  Define $\sigma_1 = [\overline{V/X}]$. By Lemma 7.13 and (69) we get

$$\begin{array}{c}\emptyset \vdash N_0 \leq T_0 \\ (\forall i \in [n])\ \emptyset \vdash N_i \leq \sigma_1 U_i\end{array}$$

  Case distinction on the form of m.

– *Case* $m = m^{\mathrm{c}}$: From (69) we get by inverting rule MTYPE-CLASS that $T_0 = C\langle\overline{T}\rangle$ with $C \neq \texttt{Object}$. By Lemma 7.10 we have $N_0 \trianglelefteq_{c} C\langle\overline{T}\rangle$. Hence, with Lemma 7.20

$$\mathsf{getmdef}^{\mathrm{c}}(m, N_0) = \langle\overline{X'}\rangle\,\overline{U'\,x'} \to U' \;\texttt{where}\; \overline{\mathcal{Q}}\,\{e''\}$$

such that $\overline{X}$ and $\overline{X'}$ as well as $\overline{U\,x}$ and $\overline{U'\,x'}$ have the same length. But then by rule DYN-INVOKE-C

$$e_0.m\langle\overline{V}\rangle(\overline{e}^n) \longrightarrow [e_0/this, \overline{e/x'}][\overline{V/X'}]e''$$

– *Case* $m = m^{\mathrm{i}}$: Then we can invert rule MTYPE-IFACE and get

$$\frac{\begin{array}{c}\texttt{interface}\; I\langle\overline{Z'}\rangle\,[\overline{Z}^{l}\;\texttt{where}\,\overline{R}]\;\texttt{where}\;\overline{P}\,\{\ldots\;\overline{rcsig}\,\} \\ rcsig_j = \texttt{receiver}\,\{\overline{m : msig}\} \\ \emptyset \Vdash \overline{T}\,\texttt{implements}\,I\langle\overline{T''}\rangle \qquad m_k = m \qquad T_j = T_0\end{array}}{\mathsf{mtype}_{\emptyset}(m, T_0) = \underbrace{[\overline{T/Z}, \overline{T''/Z'}]msig_k}_{=\langle\overline{X}^{p}\rangle\,\overline{U\,x}^{n}\to U\;\texttt{where}\;\overline{\mathcal{P}}}}\;\text{MTYPE-IFACE} \tag{71}$$
{eq:invoke-d-p1::theorem:progress}

Define $\sigma_2 = [\overline{T/Z}, \overline{T''/Z'}]$. By Lemma 7.24, we get

$$\texttt{implementation}\langle\overline{Z''}\rangle\; I\langle\overline{T'''}\rangle\;[\,\overline{M}\,]\;\texttt{where}\;\overline{Q}\;\ldots \tag{72}$$
{eq:impl-def::theorem:progress}

$$\mathsf{dom}(\sigma_3) = \overline{Z''}$$

$$\emptyset \Vdash \sigma_3\overline{Q}$$

$$\overline{T''} = \sigma_3\overline{T'''}$$

$$\emptyset \vdash N_0 \leq \sigma_3 M_j \tag{73}$$
{eq:N0-sub-Mj::theorem:progress}

$$j \in \mathsf{pos}^{+}(I) \text{ or } \emptyset \vdash T_j \leq \sigma_3 M_j \tag{74}$$
{eq:Tj-sub-Mj::theorem:progress}

$$(\forall i \neq j)\; \emptyset \vdash T_i \leq \sigma_3 M_i \tag{75}$$
{eq:Ti-sub-Mi::theorem:progress}

Assume

$$msig_k = \langle\overline{X}\rangle\,\overline{U'\,x} \to U' \;\texttt{where}\; \overline{P} \tag{76}$$
{eq:invoke-d-p2::theorem:progress}

Suppose $i \in [l]$. Then define

$$M_i^{?} = \begin{cases} \mathsf{contrib}_{Z_i}(\overline{U'}, \overline{N}) & \text{if } i \neq j \\ \mathsf{contrib}_{Z_j}(Z_j\overline{U'}, N_0\overline{N}) & \text{otherwise} \end{cases} \tag{77}$$
{eq:def-Mi::theorem:progress}

Our goal is now to prove

$$(\forall i \in [l])\; M_i^{?} = \mathsf{nil} \text{ or } M_i^{?} \trianglelefteq_{c} \sigma_3 M_i \tag{78}$$
{eq:goal1::theorem:progress}

Assume $i \in [l]$ and $M_i^{?} \neq \mathsf{nil}$. We then show $M_i^{?} \trianglelefteq_{c} \sigma_3 M_i$. First, we define

$$\mathscr{C}_i = \{N_p \mid p \in [n], U'_p = Z_i\}$$

and show that $N_p \trianglelefteq_{c} \sigma_3 M_i$ for all $N_p \in \mathscr{C}_i$. Assume $N_p \in \mathscr{C}_i$. Then $p \in [n]$ and $U'_p = Z_i$. Hence,

$$U_p = \sigma_2 U'_p = \sigma_2 Z_i = T_i$$

From (69), we then have

$$\emptyset; \emptyset \vdash e_p : \sigma_1 T_i$$

W.l.o.g., $\overline{X} \cap \mathsf{ftv}(\overline{T}) = \emptyset$, so $\sigma_1 T_i = T_i$. From (70) we have $e_p = \mathtt{new}\, N_p(\overline{w_p})$. Thus, with Lemma 7.29 we get

$$\emptyset \vdash N_p \leq T_i$$

If $i = j$ then $U'_p = Z_j$, so $j \notin \mathsf{pos}^+(I)$. With (74) and (75) we thus have $\emptyset \vdash T_i \leq \sigma_3 M_i$. Thus, by transitivity of subtyping $\emptyset \vdash N_p \leq \sigma_3 M_i$, so with Lemma 7.30

$$N_p \trianglelefteq_c \sigma_3 M_i \text{ for all } N_p \in \mathscr{C}_i \tag{79}$$ {eq:Np-sub-Mi::theorem:progress}

Now we show $M_i^? \trianglelefteq_c \sigma_3 M_i$ depending on whether or not $i = j$.

* If $i \neq j$, then, by (77) and the definition of $\mathsf{contrib}$

$$M_i^? = \bigsqcup \mathscr{C}_i$$

  The claim follows from (79) and Lemma 7.32.
* If $i = j$, then, by (77) and the definition of $\mathsf{contrib}$

$$M_i^? = \bigsqcup (\{N_0\} \cup \mathscr{C}_i)$$

  The claim follows from (79), (73), and Lemma 7.32.

This finishes the prove of (78)

We now define

$$\begin{aligned} \mathscr{M} = \{&(\sigma_4, \mathtt{implementation}\langle\overline{Z'''}\rangle\ I\langle\overline{W'}\rangle\ [\,\overline{M'}\,]\ \mathtt{where}\ \overline{Q'}\ \ldots) \\ &\mid \mathsf{dom}(\sigma_4) = \overline{Z'''}, (\forall i \in [l])\ M_i^? = \mathsf{nil} \text{ or } M_i^? \trianglelefteq_c \sigma_4 M'_i\} \end{aligned} \tag{80}$$ {eq:def-M::theorem:progress}

With (78) we have $(\sigma_3, \mathit{impl}) \in \mathscr{M}$ where impl is the implementation definition from (72). Clearly, $\mathscr{M}$ is also finite because a program has only finitely many implementation definitions. Moreover, suppose $i \in [l]$, $i \in \mathsf{disp}(I)$. Hence, either $i = j$ or there exists some argument type $U_{i'}$ with $U_{i'} = Z_i$. In any case, we have with (77) that $M_i^? \neq \mathsf{nil}$. With Lemma 7.27 we then get that there exists (σ, impl') such that

$$\mathsf{minimpl}\mathscr{M} = (\sigma, \mathit{impl}') \tag{81}$$ {eq:invoke-d-p3::theorem:progress}

Assume $\mathit{impl}' = \mathtt{implementation} \ldots \{\ldots \overline{\mathit{rcdef}}\}$ Because the underlying program is well-formed, it is easy to check that

$$\begin{aligned} \mathit{rcdef}_j &= \mathtt{receiver}\,\{\overline{\mathit{mdef}}\} \\ \mathit{mdef}_k &= \langle\overline{X'}^p\rangle\, \overline{U''\, x''}^n \to U''\ \mathtt{where}\ \overline{P'}\,\{e''\} \end{aligned} \tag{82}$$ {eq:invoke-d-p4::theorem:progress}

With (71), (76), (77), (80), (81), (82), and rule DYN-MDEF-I, we get

$$\mathsf{getmdef}^{\mathsf{i}}(m, N_0, \overline{N}) = \sigma\, \mathit{mdef}_k$$

Hence, with rule DYN-INVOKE-I and DYN-CONTEXT

$$e_0.m\langle\overline{V}\rangle(\overline{e}^n) \longrightarrow [e_0/\mathit{this}, \overline{e/x''}][\overline{V/X'}]e''$$

End case distinction on the form of m.

* *Case* rule EXP-INVOKE-S: Then

$$\frac{\begin{array}{c}\mathsf{smtype}_\emptyset(m, I\langle\overline{V}\rangle[\overline{T}]) = \langle\overline{X}^p\rangle\, \overline{U\,x}^n \to U\ \mathtt{where}\ \overline{\mathcal{P}} \qquad (\forall i)\ \emptyset;\emptyset \vdash e_i : [\overline{W/X}]U_i \\ \emptyset \Vdash [\overline{W/X}]\overline{\mathcal{P}} \qquad 1 \notin \mathsf{pos}^+(I) \text{ or } (\exists i)\ \emptyset \Vdash T_i\, \mathsf{mono} \qquad \emptyset \vdash \overline{T}, \overline{W}\ \mathsf{ok}\end{array}}{\emptyset;\emptyset \vdash \underbrace{I\langle\overline{V}\rangle[\overline{T}^l].m\langle\overline{W}\rangle(\overline{e})}_{=e} : \underbrace{[\overline{V/X}]U}_{=T}}\ \textsc{exp-invoke-s} \tag{83}$$ {eq:invoke-s::theorem:progress}

We now apply the I.H. to $\emptyset; \emptyset \vdash e_i : [\overline{W/X}]U_i$, for $i = 1, \ldots, n$. As in the case for rule EXP-INVOKE, the only interesting case is the one where

$$(\forall i)\ e_i = v_i = \texttt{new}\ N_i(\overline{w_i})$$

Define $\sigma_1 = [\overline{W/X}]$. With Lemma 7.29 we have

$$(\forall i)\ \emptyset \vdash N_i \leq \sigma_1 U_i$$

Inverting rule MTYPE-STATIC yields

$$\frac{\begin{array}{c}\texttt{interface}\ I\langle \overline{Z'}^{l}\rangle\ [\overline{Z\ \texttt{where}\ \overline{R}}]\ \texttt{where}\ \overline{Q}\ \{\ \overline{m : \texttt{static}\ msig}\ \ldots\} \\ \emptyset \Vdash \overline{T}\ \texttt{implements}\ I\langle\overline{V}\rangle \qquad m = m_k\end{array}}{\mathsf{smtype}_{\emptyset}(m, I\langle\overline{V}\rangle[\overline{T}]) = \underbrace{[\overline{V/Z'}, \overline{T/Z}]}_{=\sigma_2}\ msig_k}\ \text{MTYPE-STATIC}$$

Assume $1 \in \mathsf{pos}^{+}(I)$. By (83) we then have $\emptyset \Vdash T_j\ \texttt{mono}$ for some j. Thus, with Lemma 7.12 we have $T_j = M$ for some M. We now use Lemma 7.24 for the case $1 \in \mathsf{pos}^{+}(I)$ and Lemma 7.25 for the case $1 \notin \mathsf{pos}^{+}(I)$ and get

$$\begin{array}{c} impl = \texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{V'}\rangle\ [\ \overline{N}^{l}\ ]\ \texttt{where}\ \overline{Q'}\ \ldots \\ \mathsf{dom}(\sigma_3) = \overline{Y} \\ \emptyset \Vdash \sigma_3 \overline{Q'} \\ \overline{V} = \sigma_3 \overline{V'} \\ (\forall i \in [l])\ \emptyset \vdash T_i \leq \sigma_3 N_i \end{array}$$

With Lemma 7.10 we then get for all $i \in [l]$

$$N_i = \texttt{Object}\ \text{or}\ T_i = M_i\ \text{for some}\ M_i\ \text{with}\ M_i \trianglelefteq_c \sigma_3 N_i$$

Now define

$$\begin{array}{rl}\mathscr{M} = \{(\sigma_4, \texttt{implementation}\langle\overline{Y'}\rangle\ I\langle\overline{V''}\rangle\ [\ \overline{N'}^{l}\ ]\ \texttt{where}\ \overline{Q''}\ \ldots) \\ \mid \mathsf{dom}(\sigma_4) = \overline{Y'}, (\forall i \in [l])\ N'_i = \texttt{Object}\ \text{or}\ T_i \trianglelefteq_c \sigma_4 N_i\}\end{array}$$

Clearly, $(\sigma_3, impl) \in \mathscr{M}$. Moreover, $\mathscr{M}$ is finite because programs contain only finitely many implementation definitions. Hence, by Lemma 7.28 we know that there exists $(\sigma, impl')$ such that

$$\mathsf{minimpl}\mathscr{M} = (\sigma, impl')$$

Suppose that $\overline{\texttt{static}\ mdef}$ are the static methods of $impl'$. Because the underlying program is well-typed, we know that $mdef_k = \langle\overline{X'}^{p}\rangle\ \overline{U'\ x'}^{n} \to U'\ \texttt{where}\ \overline{P'}\ \{e''\}$. Hence, we have

$$\mathsf{getsmdef}(m, I\langle\overline{V}\rangle[\overline{T}]) = \sigma\, mdef_k$$

by rule DYN-MDEF-S and so

$$I\langle\overline{V}\rangle[\overline{T}].m\langle\overline{W}\rangle(\overline{e}) \longrightarrow [\overline{e/x'}][\overline{W/X'}]e''$$

by rule DYN-INVOKE-S and rule DYN-CONTEXT.

- *Case* rule EXP-NEW: Then $e = \texttt{new}\ N(\overline{e}^{n})$ and $(\forall i)\ \emptyset; \emptyset \vdash e_i : T_i$. Applying the I.H. yields three possibilities:
  - All e_i are values. Then e is a value.

– The first m expressions are values ($m < n$) and $e_{m+1} \longrightarrow e'_{m+1}$. Then $e \longrightarrow \texttt{new}\, N(e_1, \ldots, e_m, e'_{m+1}, e_{m+2}, \ldots, e_n)$.
– The first m expressions are values ($m < n$) and and e_{m+1} is stuck on a bad cast. Then e is stuck on a bad cast as well.

- *Case* rule EXP-CAST: Then

$$\frac{\emptyset \vdash N \ \mathsf{ok} \qquad \emptyset; \emptyset \vdash e_0 : T}{\emptyset; \emptyset \vdash (N)\, e_0 : N}\ \textsc{exp-cast}$$

with $e = (N)\, e_0$. Applying the I.H. leaves us with three possibilities:

– e_0 is a value. Then $e_0 = \texttt{new}\, M(\overline{v})$. If $M \trianglelefteq_c N$ then $e \longrightarrow e_0$ by rules DYN-CAST and DYN-CONTEXT. Otherwise, e is stuck on a bad cast.
– $e_0 \longrightarrow e'_0$. Then $e \longrightarrow (N)\, e'_0$ by rule DYN-CONTEXT.
– e_0 is stuck on a bad cast. Then e is also stuck on a bad cast.

- *Case* rule EXP-SUBSUME: In this case, the claim follows directly from the I.H.

End case distinction on the last rule of the derivation of $\emptyset; \emptyset \vdash e : T$. □

Lemma 7.35. *If* $\mathsf{fields}(N) = \overline{T\, f}$ *and* $\mathsf{fields}(N) = \overline{U\, g}$ *then* $\overline{T} = \overline{U}$ *and* $\overline{f} = \overline{g}$.

PROOF. Straightforward induction on the derivation of $\mathsf{fields}(N) = \overline{T\, f}$. □

Lemma 7.36. *If* $N_1 \sqcup N_2 = M$ *then* $N_i \trianglelefteq_c M$ *for* $i = 1, 2$.

PROOF. Straightforward induction on the derivation of $N_1 \sqcup N_2 = M$. □

Lemma 7.37. *If* $N \in \mathscr{N}$ *and* $M = \bigsqcup \mathscr{N}$ *then* $N \trianglelefteq_c M$.

PROOF. Straightforward induction on the derivation of $M = \bigsqcup \mathscr{N}$, making use of Lemma 7.36. □

Lemma 7.38 (Well-formedness for subterms)**.**

(*i*) *If* $\Delta \vdash [U/X]T$ ok *and* $X \in \mathsf{ftv}(T)$ *then* $\Delta \vdash U$ ok.

(*ii*) *If* $\Delta \vdash [U/X]\mathcal{P}$ ok *and* $X \in \mathsf{ftv}(\mathcal{P})$ *then* $\Delta \vdash U$ ok.

PROOF. We prove both parts by routine inductions on the derivations given. □

Lemma 7.39. *If* $\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}^l]\ \ldots$ *and* $M_i^? \neq \mathsf{nil}$ *for all* $i \in \mathsf{disp}(I)$ *and, for all* $i \in [l]$ *with* $M_i^? \neq \mathsf{nil}$, $\Delta \vdash M_i^?$ ok *and* $M_i^? \trianglelefteq_c [\overline{U/X}]N_i$, *then* $\Delta \vdash \overline{U}$ ok.

PROOF. Suppose $i \in [l]$ such that $M_i^? \neq \mathsf{nil}$. Then we get with Lemma 7.16 that $\Delta \vdash [\overline{U/X}]N_i$ ok. By Lemma 7.38 we then know that $\Delta \vdash U_j$ ok for all j with $X_j \in \mathsf{ftv}(N_i)$. Moreover, by criterion WF-IMPL-1 we have that $\overline{X} \subseteq \mathsf{ftv}\{N_i \mid i \in \mathsf{disp}(I)\}$. Hence, $\Delta \vdash \overline{U}$ ok. □

Lemma 7.40. *If* $\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}^l]\ \ldots$ *and for all* $i \in [l]$ *either* $N_i = \texttt{Object}$ *or* $M_i \trianglelefteq_c [\overline{U/X}]N_i$ *for some* M_i *with* $\emptyset \vdash M_i$ ok, *then* $\Delta \vdash \overline{U}$ ok.

PROOF. The proof is similar to that of Lemma 7.39. □

Lemma 7.41. *If* $\bigsqcup \mathscr{N} = M$ *and* $\Delta \vdash N$ ok *for some* $N \in \mathscr{N}$, *then* $\Delta \vdash M$ ok.

PROOF. From Lemma 7.37, we have $N \trianglelefteq_c M$. Because $\Delta \vdash N$ ok we then have $\Delta \vdash M$ ok by Lemma 7.16. □

Theorem 7.42 (Preservation for top-level evaluation). *If $\emptyset; \emptyset \vdash e : T$ and $e \longmapsto e'$ then $\emptyset; \emptyset \vdash e' : T$.*

Proof. By induction on the derivation of $\emptyset; \emptyset \vdash e : T$.
Case distinction on the last rule of the derivation of $\emptyset; \emptyset \vdash e : T$.

- *Case* rule exp-var: Impossible.
- *Case* rule exp-field: Then

$$\frac{\emptyset; \emptyset \vdash e_0 : C\langle\overline{T}\rangle \qquad \texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ M\ \texttt{where}\ \overline{P}\,\{\,\overline{U\,f}\ldots\}}{\emptyset; \emptyset \vdash e_0.f_j : [\overline{T/X}]U_j}\ \text{exp-field}$$

with $T = [\overline{T/X}]U_j$ and $e = e_0.f_j$. From $e \longmapsto e'$ we get

$$\begin{aligned} e_0 &= \texttt{new}\,N(\overline{v}) \\ \mathsf{fields}(N) &= \overline{V\,f'} \\ e' &= v_i \\ f'_i &= f_j \end{aligned}$$

We have by Lemma 7.13, inspection of the expression typing rules, Lemma 7.30, and Lemma 7.35 that

$$\frac{\emptyset; \emptyset \vdash N\ \mathsf{ok} \qquad \mathsf{fields}(N) = \overline{V\,f'} \qquad (\forall i)\ \emptyset; \emptyset \vdash v_i : V_i}{\emptyset; \emptyset \vdash \texttt{new}\,N(\overline{v}) : N}\ \text{exp-new}$$

such that $N \trianglelefteq_c C\langle\overline{T}\rangle$. From Lemma 7.21 we get $V_i = [\overline{T/X}]U_j$, so $\emptyset; \emptyset \vdash e' : T$ as required.

- *Case* rule exp-invoke: Then

$$\frac{\begin{array}{c}\emptyset; \emptyset \vdash v_0 : T_0 \qquad \mathsf{mtype}_{\emptyset}(m, T_0) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}} \\ (\forall i \in [n])\ \emptyset; \emptyset \vdash v_i : [\overline{V/X}]U_i \qquad \emptyset \Vdash [\overline{V/X}]\overline{\mathcal{P}} \qquad \emptyset \vdash \overline{V}\ \mathsf{ok}\end{array}}{\emptyset; \emptyset \vdash \underbrace{v_0.m\langle\overline{V}\rangle(\overline{v}^n)}_{=e} : \underbrace{[\overline{V/X}]U}_{=T}}\ \text{exp-invoke} \tag{84}$$

{eq:exp-invoke-d::theorem:preservation-stepsto}

Case distinction on the rule used to reduce e.

 – *Case* rule dyn-invoke-c: Then

$$\begin{aligned} v_0 &= \texttt{new}\,N(\overline{w}) \\ \mathsf{getmdef}^{c}(m, N) &= \langle\overline{X'}\rangle\,\overline{U'\,x'} \to U'\ \texttt{where}\ \overline{\mathcal{P}'}\,\{e''\} \\ e' &= [v_0/\texttt{this}, \overline{v/x}][\overline{V/X'}]e'' \\ m &= m^{c} \end{aligned}$$

By definition of mtype, we know that $T_0 = N'$ for some N'. By Lemma 7.29 and Lemma 7.30 we get

$$\begin{aligned} N &\trianglelefteq_c N' \\ \emptyset &\vdash N\ \mathsf{ok} \end{aligned}$$

We now get with Lemma 7.19 that

$$\begin{aligned} \overline{X} &= \overline{X'} \\ \overline{x} &= \overline{x'} \\ \emptyset; \texttt{this} : N, \overline{x : [\overline{V/X}]U} &\vdash [\overline{V/X}]e : [\overline{V/X}]U \end{aligned}$$

Possibly repeated applications of Lemma 7.8 yield

$$\emptyset; \emptyset \vdash e' : T$$

– *Case* rule DYN-INVOKE-I: Then $m = m^{\mathrm{i}}$ and $e' = [v_0/\mathtt{this}, \overline{v/x}][\overline{V/X}]\sigma_1 e''$ and

$$\frac{\begin{array}{c}\mathtt{interface}\ I\langle\overline{Z'}\rangle\ [\overline{Z}^{l}\ \mathtt{where}\ \overline{R}]\ \mathtt{where}\ \overline{Q'}\ \{\ldots\ \overline{rcsig}\ \} \\ rcsig_j = \mathtt{receiver}\,\{\overline{m : msig}\} \qquad m = m_k \qquad msig_k = \langle\overline{X''}\rangle\ \overline{W\,x''} \to W\ \mathtt{where}\ \overline{Q} \\ (\forall i \in [l], i \neq j)\ \mathsf{contrib}_{Z_i}(\overline{W}, \overline{N}) = M_i^{?} \qquad \mathsf{contrib}_{Z_j}(Z_j\overline{W}, N_0\overline{N}) = M_j^{?} \\ (\sigma_1, \mathtt{implementation}\langle\overline{Z''}\rangle\ I\langle\overline{W''}\rangle\ [\overline{M'}]\ \mathtt{where}\ \overline{Q''}\ \{\ldots\ \overline{rcdef}\ \}) = \mathsf{minimpl}\,\mathscr{M} \\ rcdef_j = \mathtt{receiver}\,\{\overline{mdef}\}\end{array}}{\mathsf{getmdef}^{\mathrm{i}}(m, N_0, \overline{N}) = \sigma_1\, mdef_k}\ \text{DYN-MDEF-I} \tag{85}$$

{eq:getmdef-i::theorem:preservation-stepsto}

where

$$mdef_k = \langle\overline{X'}\rangle\ \overline{U'\,x'} \to U'\ \mathtt{where}\ \overline{P'}\ \{e''\}$$

$$(\forall i \in \{0, \ldots, n\})\ v_i = \mathtt{new}\, N_i(\overline{w_i}) \tag{86}$$

{eq:def-vi::theorem:preservation-stepsto}

$$\begin{aligned}\mathscr{M} = \{&(\sigma, \mathtt{implementation}\langle\overline{Z''}\rangle\ I\langle\overline{W''}\rangle\ [\overline{M'}]\ \ldots) \\ &\mid \mathsf{dom}(\sigma) = \overline{Z''}, (\forall i \in [l])\ M_i^{?} = \mathsf{nil}\ \text{or}\ M_i^{?} \trianglelefteq_{\mathrm{c}} \sigma M_i'\}\end{aligned}$$

By definition of mtype and criterion WF-PROG-1, we have from (84) that

$$\frac{\begin{array}{c}\mathtt{interface}\ I\langle\overline{Z'}\rangle\ [\overline{Z}^{l}\ \mathtt{where}\ \overline{R}]\ \mathtt{where}\ \overline{Q'}\ \{\ldots\ \overline{rcsig}\ \} \\ rcsig_j = \mathtt{receiver}\,\{\overline{m : msig}\} \\ m = m_k \qquad \emptyset \Vdash \overline{T}\ \mathtt{implements}\, I\langle\overline{T''}\rangle \qquad T_j = T_0\end{array}}{\mathsf{mtype}_{\emptyset}(m, T_0) = \underbrace{[\overline{T''/Z'}, \overline{T/Z}]msig_k}_{=\langle\overline{X}\rangle\ \overline{U\,x} \to U\ \mathtt{where}\ \overline{\mathcal{P}}}}\ \text{MTYPE-IFACE} \tag{87}$$

{eq:mtype::theorem:preservation-stepsto}

With $\sigma_2 = [\overline{T''/Z'}, \overline{T/Z}]$ we then get

$$\overline{X} = \overline{X''} \tag{88}$$

{eq:X=X''::theorem:preservation-stepsto}

$$\overline{x} = \overline{x''} \tag{89}$$

{eq:x=x''::theorem:preservation-stepsto}

$$\sigma_2(\overline{W}, W, \overline{Q}) = \overline{U}, U, \overline{\mathcal{P}} \tag{90}$$

{eq:wq=up::theorem:preservation-stepsto}

The underlying program is well-typed, so we have

$$\overline{Q''}, \overline{Z''}; \mathtt{this} : M_j' \vdash rcdef_j\ \mathsf{implements}\ \underbrace{[\overline{W''/Z'}, \overline{M'/Z}]}_{=\sigma_3}\, rcsig_j$$

This especially implies

$$\overline{Q''}, \overline{Z''}; \mathtt{this} : M_j' \vdash mdef_k\ \mathsf{implements}\ \sigma_3\, msig_k$$

which in turn implies

$$\underbrace{\overline{Q''}, \overline{Z''}, \overline{P'}, \overline{X'}}_{=\Delta} \vdash \overline{U'}, U', \overline{P'}\ \mathsf{ok} \tag{91}$$

{eq:def-delta::theorem:preservation-stepsto}

$$\Delta; \underbrace{\mathtt{this} : M_j', \overline{x' : U'}}_{=\Gamma} \vdash e'' : U' \tag{92}$$

{eq:type-e''::theorem:preservation-stepsto}

$$\overline{X'} = \overline{X''} \tag{93}$$

{eq:X'=X''::theorem:preservation-stepsto}

$$\overline{x'} = \overline{x''} \tag{94}$$

{eq:x'=x''::theorem:preservation-stepsto}

$$\overline{U'} = \sigma_3 \overline{W} \tag{95}$$

{eq:u'=w::theorem:preservation-stepsto}

$$\overline{P'} = \sigma_3 \overline{Q} \tag{96}$$

{eq:p'=q::theorem:preservation-stepsto}

$$\Delta \vdash U' \leq \sigma_3 W \tag{97}$$

{eq:U'-st-w::theorem:preservation-stepsto}

By (84) we get $\emptyset; \emptyset \vdash v_0 : T_0$, so with (86) and Lemma 7.29:

$$\emptyset \vdash N_0 \leq T_0 \tag{98}$$
{eq:N0-st-T0::theorem:preservation-stepsto}

Using (87) we get $\emptyset \Vdash \overline{T}\ \texttt{implements}\ I\langle\overline{T''}\rangle$ with $T_j = T_0$. Applying Lemma 7.24 yields

$$
\begin{gathered}
\mathit{impl} = \texttt{implementation}\langle\overline{Z_3}\rangle\ I\langle\overline{W_3}\rangle\ [\,\overline{M_3}\,]\ \texttt{where}\ \overline{Q_3}\ \{\ldots\ \overline{\mathit{rcdef}'}\ \} \\
\emptyset \Vdash \sigma_4\overline{Q_3} \qquad (99) \\
\mathsf{dom}(\sigma_4) = \overline{Z_3} \\
\overline{T''} = \sigma_4\overline{W_3} \qquad (100) \\
\emptyset \vdash N_0 \leq \sigma_4 M_{3j} \qquad (101) \\
\text{if } j \notin \mathsf{pos}^+(I) \text{ then } \emptyset \vdash T_j \leq \sigma_4 M_{3j} \text{ with} \\
T_j \neq \sigma_4 M_{3j} \text{ implying } j \in \mathsf{pos}^-(I) \qquad (102) \\
(\forall i \neq j)\ \emptyset \vdash T_i \leq \sigma_4 M_{3i} \text{ with } T_i \neq \sigma_4 M_{3i} \text{ implying } i \in \mathsf{pos}^-(I) \qquad (103) \\
\text{if } j \in \mathsf{pos}^+(I) \text{ and } j \notin \mathsf{pos}^-(I) \text{ and } T_j \neq \sigma_4 M_{3j} \text{ then} \\
\overline{T} = T_j = J\langle\overline{W_4}\rangle \text{ and } J\langle\overline{W_4}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{W_3}\rangle \text{ and } 1 \in \mathsf{pos}^+(J) \qquad (104)
\end{gathered}
$$
{eq:entails-q3::theorem:preservation-stepsto}
{eq:t''=w3::theorem:preservation-stepsto}
{eq:N0-st-M3j::theorem:preservation-stepsto}
{eq:impl1::theorem:preservation-stepsto}
{eq:impl2::theorem:preservation-stepsto}
{eq:impl3::theorem:preservation-stepsto}

We now show that $(\sigma_4, \mathit{impl}) \in \mathscr{M}$. To do so, we prove that $(\forall i \in [l]) M_i^? = \mathsf{nil}$ or $M_i^? \trianglelefteq_{\mathrm{c}} \sigma_4 M_{3i}$. Suppose $i \in [l]$ and assume $M_i^? \neq \mathsf{nil}$. By definition of $M_i^?$ in (85) and by Lemma 7.32, it suffices to show that $N_p \trianglelefteq_{\mathrm{c}} \sigma_4 M_{3i}$ for all $p \in [n]$ with $W_p = Z_i$, and that $N_0 \trianglelefteq_{\mathrm{c}} \sigma_4 M_{3j}$. The latter follows directly from (101) and Lemma 7.30. Now assume $p \in [n]$ with $W_p = Z_i$. Then

$$\sigma_2 W_p = T_i$$

From (84) we have $\emptyset; \emptyset \vdash v_p : [\overline{V/X}]U_p$, so with (90)

$$\emptyset; \emptyset \vdash v_p : [\overline{V/X}]T_i$$

W.l.o.g., $\overline{X} \cap \mathsf{ftv}(T_i) = \emptyset$, so $[\overline{V/X}]T_i = T_i$. Thus, with (86) and Lemma 7.29

$$\emptyset \vdash N_p \leq T_i$$

Because $W_p = Z_i$, we have $i \notin \mathsf{pos}^+(I)$. Hence, we get from (102) and (103) that $\emptyset \vdash T_i \leq \sigma_4 M_{3i}$. By transitivity of subtyping and Lemma 7.30 we then get $N_p \trianglelefteq_{\mathrm{c}} \sigma_4 M_{3i}$ as required. We now have established the fact that

$$(\sigma_4, \mathit{impl}) \in \mathscr{M}$$

From (85) and the definition of $\mathsf{minimpl}$, we get that

$$(\forall i \in [l])\ \sigma_1 M'_i \trianglelefteq_{\mathrm{c}} \sigma_4 M_{3i} \tag{105}$$
{eq:M'=M3i::theorem:preservation-stepsto}

We then get from (99) and Criterion Wf-Prog-4 that

$$\emptyset \Vdash \sigma_1\overline{Q''} \tag{106}$$
{eq:entails-subst1-q''::theorem:preservation-stepsto}

By criterion Wf-Prog-2 we get $\sigma_1\overline{W''} = \sigma_4\overline{W_3}$, so with (100)

$$\sigma_1\overline{W''} = \overline{T''} \tag{107}$$
{eq:w''=t''::theorem:preservation-stepsto}

By criterion Wf-Iface-4 we have $\overline{Z} \cap \mathsf{ftv}(\overline{Q}) = \emptyset$. Then $\mathsf{ftv}(\overline{Q}) \subseteq \overline{Z'}$ as the underlying program is well-typed. W.l.o.g., $\overline{Z''} \cap \mathsf{ftv}(\overline{Q}) = \emptyset$, so

$$\sigma_1\sigma_3\overline{Q} = \sigma_1[\overline{W''/Z'}]\overline{Q} = [\overline{\sigma_1 W''/Z'}]\overline{Q} = [\overline{T''/Z'}]\overline{Q} = \sigma_2\overline{Q}$$

From (84) and (90) we get $\emptyset \Vdash [\overline{V/X}]\sigma_2\overline{Q}$. Thus,

$$\emptyset \Vdash [\overline{V/X}]\sigma_1\sigma_3\overline{Q} \tag{108}$$ {eq:entails-q::theorem:preservation-stepsto}

We have $v_0 = \texttt{new}\, N_0(\overline{w_0})$ by (86). By (85), the definition of $\mathsf{contrib}$, and Lemma 7.37: $N_0 \trianglelefteq_c M_j^?$. Moreover, $M_j^? \trianglelefteq_c \sigma_1 M_j'$ by definition of $\mathscr{M}$ and $\mathsf{minimpl}$. Then with EXP-SUBSUME $\emptyset;\emptyset \vdash v_0 : \sigma_1 M_j'$. We have $\emptyset \vdash \overline{V}\ \mathsf{ok}$ by (84) so $\emptyset;\emptyset \vdash [\overline{V/X}]v_0 : [\overline{V/X}]\sigma_1 M_j'$ by Lemma 7.7. W.l.o.g., $\overline{X} \cap \mathsf{ftv}(v_0) = \emptyset$, so

$$\emptyset;\emptyset \vdash v_0 : [\overline{V/X}]\sigma_1 M_j' \tag{109}$$ {eq:v0-Mj'::theorem:preservation-stepsto}

Next, we prove that $\emptyset;\emptyset \vdash v_i : [\overline{V/X}]\sigma_1 U_i'$ for all $i \in [n]$. Assume $i \in [n]$. By criterion WF-IFACE-4 we have either $\overline{Z} \cap \mathsf{ftv}(W_i) = \emptyset$ or $W_i \in \overline{Z}$. Because the underlying program is well-typed, we have $\mathsf{ftv}(W_i) \subseteq \{\overline{Z}, \overline{Z'}\}$. W.l.o.g., $\overline{Z''} \cap \mathsf{ftv}(W_i) = \emptyset$.

* Assume $\overline{Z} \cap \mathsf{ftv}(W_i) = \emptyset$. Then

$$\sigma_1\sigma_3 W_i = \sigma_1[\overline{W''/Z'}]W_i = [\overline{\sigma_1 W''/Z'}]W_i \overset{(107)}{=} [\overline{T''/Z'}]W_i = \sigma_2 W_i$$

  Hence,

$$U_i \overset{(90)}{=} \sigma_2 W_i = \sigma_1\sigma_3 W_i \overset{(95)}{=} \sigma_1 U_i'$$

  From (84) we have $\emptyset;\emptyset \vdash v_i : [\overline{V/X}]U_i$. Thus, $\emptyset;\emptyset \vdash v_i : [\overline{V/X}]\sigma_1 U_i'$.

* Assume $W_i = Z_k$ for some $k \in [l]$. We have $v_i = \texttt{new}\, N_i(\overline{w_i})$ by (86). By (85), the definition of $\mathsf{contrib}$, and Lemma 7.37: $M_k^? \neq \mathsf{nil}$ and $N_i \trianglelefteq_c M_k^?$. Moreover, $M_k^? \trianglelefteq_c \sigma_1 M_k'$ by definition of $\mathscr{M}$. By rule EXP-NEW, Lemma 6.2, Lemma 7.18, and rule EXP-SUBSUME we then have $\emptyset;\emptyset \vdash v_i : \sigma_1 M_k'$. We also have

$$\sigma_1 M_k' = \sigma_1\sigma_3 Z_k = \sigma_1\sigma_3 W_i \overset{(95)}{=} \sigma_1 U_i'$$

  We have $\emptyset \vdash \overline{V}\ \mathsf{ok}$ by (84) so $\emptyset;\emptyset \vdash [\overline{V/X}]v_i : [\overline{V/X}]\sigma_1 U_i'$ by Lemma 7.7. W.l.o.g., $\overline{X} \cap \mathsf{ftv}(v_0) = \emptyset$, so $\emptyset;\emptyset \vdash v_i : [\overline{V/X}]\sigma_1 U_i'$.

This finishes the proof of

$$(\forall i \in [n])\ \emptyset;\emptyset \vdash v_i : [\overline{V/X}]\sigma_1 U_i' \tag{110}$$ {eq:vi-Ui::theorem:preservation-stepsto}

Next, we prove $\emptyset \vdash \sigma_1\sigma_3 W \leq \sigma_2 W$. Note that $\mathsf{ftv}(W) \subseteq \{\overline{Z}, \overline{Z'}\}$ because the underlying program is well-typed. W.l.o.g., $\overline{Z''} \cap \mathsf{ftv}(W) = \emptyset$.

Case distinction on whether or not $\overline{Z} \cap \mathsf{ftv}(W) = \emptyset$.

* *Case* $\overline{Z} \cap \mathsf{ftv}(W) = \emptyset$: Then

$$\sigma_1\sigma_3 W = \sigma_1[\overline{W''/Z'}]W = [\overline{\sigma_1 W''/Z'}]W \overset{(107)}{=} [\overline{T''/Z'}]W = \sigma_2 W$$

  By reflexivity of subtyping then

$$\emptyset \vdash \sigma_1\sigma_3 W \leq \sigma_2 W$$

* *Case* $\overline{Z} \cap \mathsf{ftv}(W) \neq \emptyset$: By criterion WF-IFACE-4 then $W = Z_k$ for some $k \in [l]$. Then

$$k \notin \mathsf{pos}^-(I) \tag{111}$$ {eq:k-not-minus::theorem:preservation-stepsto}

We first concentrate on the case where $k \neq j$ or $j \notin \mathsf{pos}^+(I)$ or $\sigma_4 M_{3k} = T_k$. Then we have

$$\sigma_1\sigma_3 W = \sigma_1\sigma_3 Z_k = \sigma_1 M'_k \overset{(105)}{\trianglelefteq_c} \sigma_4 M_{3k} \overset{(102)\text{ or }(103)\text{ or assumption}}{=} T_k \overset{\text{def. of }\sigma_2}{=} \sigma_2 Z_k = \sigma_2 W$$

Thus, we get

$$\emptyset \vdash \sigma_1\sigma_3 W \leq \sigma_2 W$$

by Lemma 7.18.

Now we consider the case $k = j$ and $j \in \mathsf{pos}^+(I)$ and $\sigma_4 M_{3k} \neq T_k$. From (104) we get

$$j = k = l = 1 \quad (112) \quad \{\texttt{eq:jk1::theorem:preservation-stepsto}\}$$

$$\overline{T} = T_j = J\langle\overline{W_4}\rangle \quad (113) \quad \{\texttt{eq:Tj=IW4::theorem:preservation-stepsto}\}$$

$$J\langle\overline{W_4}\rangle \trianglelefteq_i I\langle\overline{W_3}\rangle \quad (114) \quad \{\texttt{eq:J-sub-I::theorem:preservation-stepsto}\}$$

$$1 \in \mathsf{pos}^+(J) \quad (115) \quad \{\texttt{eq:pos+J::theorem:preservation-stepsto}\}$$

With (98) and (87) we then get $\emptyset \vdash N_0 \leq J\langle\overline{W_4}\rangle$. Applying Lemma 7.10 yields

$$\texttt{implementation}\langle\overline{Z_4}\rangle\ J\langle\overline{W'_4}\rangle\ [\,N'_0\,]\ \texttt{where}\ \overline{Q_4}\ \ldots \quad (116) \quad \{\texttt{eq:implJW4::theorem:preservation-stepsto}\}$$

$$\mathsf{dom}(\tau) = \overline{Z_4}$$

$$\emptyset \Vdash \tau\overline{Q_4} \quad (117) \quad \{\texttt{eq:entailsQ4::theorem:preservation-stepsto}\}$$

$$\tau\overline{W'_4} = \overline{W_4} \quad (118) \quad \{\texttt{eq:W4=W4::theorem:preservation-stepsto}\}$$

$$N_0 \trianglelefteq_c \tau N'_0 \quad (119) \quad \{\texttt{eq:N0-extends-N0'::theorem:preservation-stepsto}\}$$

With (114) and Lemma 6.26 we get

$$\tau N'_0\ \texttt{implements}\ I\langle\overline{W_3}\rangle \in \mathsf{sup}(\tau N'_0\ \texttt{implements}\ J\langle\overline{W_4}\rangle)$$

With Lemma 6.25 and (118) we get the existence of N''_0 and $I\langle\overline{W'_3}\rangle$ such that

$$N''_0\ \texttt{implements}\ I\langle\overline{W'_3}\rangle \in \mathsf{sup}(N'_0\ \texttt{implements}\ J\langle\overline{W'_4}\rangle)$$

$$\tau N''_0 = \tau N'_0 \quad (120) \quad \{\texttt{eq:N0''=N0'::theorem:preservation-stepsto}\}$$

$$\tau I\langle\overline{W'_3}\rangle = I\langle\overline{W_3}\rangle$$

Now by criterion Wf-Impl-2, (111), and (112)

$$\mathit{impl}' = \texttt{implementation}\langle\overline{Z_5}\rangle\ I\langle\overline{W''_3}\rangle\ [\,N'''_0\,]\ \ldots$$

$$\mathsf{dom}(\tau') = \overline{Z_5}$$

$$N''_0 = \tau' N'''_0 \quad (121) \quad \{\texttt{eq:N0''=N0'''::theorem:preservation-stepsto}\}$$

$$I\langle\overline{W'_3}\rangle = \tau' I\langle\overline{W''_3}\rangle$$

With (120) we then get $\tau N'_0 = \tau\tau' N'''_0$. Hence, with (119)

$$N_0 \trianglelefteq_c \tau\tau' N'''_0$$

From (115) it is easy to see that $Z_j \notin \mathsf{ftv}(\overline{W})$. Thus, from (85) and the definition of $\mathsf{contrib}$, we have $M^?_j = N_0$. With (112) and the definition of $\mathscr{M}$ we then have

$$(\tau\tau', \mathit{impl}') \in \mathscr{M}$$

From (85) and the definition of $\mathsf{minimpl}$ we then have

$$\sigma_1 M'_j \trianglelefteq_c \tau\tau' N'''_0$$

From (116), (117), (118), and rule ENT-IMPL, we have $\emptyset \Vdash \tau N'_0 \ \mathtt{implements}\ J\langle\overline{W_4}\rangle$. Hence, with rule SUB-IMPL then $\emptyset \vdash \tau N'_0 \leq J\langle\overline{W_4}\rangle$. With (120) and (121): $\tau N'_0 = \tau\tau' N'''_0$, and with (113): $J\langle\overline{W_4}\rangle = T_j$. With Lemma 7.18, transitivity of subtyping, and (112) we then have

$$\emptyset \vdash \sigma_1 M'_k \leq T_k$$

Moreover, we have

$$\sigma_1\sigma_3 W = \sigma_1\sigma_3 Z_k = \sigma_1 M'_k$$
$$T_k \overset{\text{def. of } \sigma_2}{=} \sigma_2 Z_k = \sigma_2 W$$

Thus, we get

$$\emptyset \vdash \sigma_1\sigma_3 W \leq \sigma_2 W$$

End case distinction on whether or not $\overline{Z} \cap \mathsf{ftv}(W) = \emptyset$.
We now have proved $\emptyset \vdash \sigma_1\sigma_3 W \leq \sigma_2 W$. Using Lemma 7.1 we conclude

$$\emptyset \vdash [\overline{V/X}]\sigma_1\sigma_3 W \leq [\overline{V/X}]\sigma_2 W \tag{122}$$ {eq:W-sub-W::theorem:preservation-stepsto}

W.l.o.g., $\mathsf{ftv}(\sigma_1\overline{Q''}) \cap \overline{X} = \emptyset$, so with (106): $\emptyset \Vdash [\overline{V/X}]\sigma_1\overline{Q''}$. From (108) and (96) we get $\emptyset \Vdash [\overline{V/X}]\sigma_1\overline{P'}$. Hence, with (91):

$$\emptyset \Vdash [\overline{V/X}]\sigma_1\Delta \tag{123}$$ {eq:entails-tenv::theorem:preservation-stepsto}

Assume $\sigma_1 = [\overline{V'/Z''}]$. W.l.o.g., $\mathsf{ftv}(\overline{V'}) \cap \overline{X} = \emptyset$. With (88) and (93) then $[\overline{V/X}]\sigma_1 = [\overline{V/X'}, \overline{V'/Z''}]$. Hence, with (91):

$$\mathsf{dom}(\Delta) \setminus \mathsf{dom}([\overline{V/X}]\sigma_1) = \emptyset$$

From (84) we have $\emptyset \vdash \overline{V}\ \mathsf{ok}$. From Lemma 7.29, (84), and (86) we get $\emptyset \vdash N_i\ \mathsf{ok}$ for all $i = 0, \ldots, n$. By definition of $\mathsf{contrib}$ and Lemma 7.41 we then get $\emptyset \vdash M^?_i\ \mathsf{ok}$ unless $M^?_i = \mathsf{nil}$. Moreover, by definition of $\mathsf{contrib}$ and disp, we get $M^?_i \neq \mathsf{nil}$ for all $i \in \mathsf{disp}(I)$. Hence, with (85), the definition of $\mathscr{M}$, and Lemma 7.39 we get $\emptyset \vdash \sigma_1 X\ \mathsf{ok}$ for all $X \in \mathsf{dom}(\sigma_1)$. Thus,

$$\emptyset \vdash [\overline{V/X}]\sigma_1 Z \text{ for all } Z \in \mathsf{dom}([\overline{V/X}]\sigma_1)$$

We now get with (92) and Lemma 7.7 that

$$\emptyset; [\overline{V/X}]\sigma_1\Gamma \vdash [\overline{V/X}]\sigma_1 e'' : [\overline{V/X}]\sigma_1 U'$$

We have with (89), (94), and (92) that $\Gamma = \mathtt{this} : M'_j, \overline{x : U'}$. Thus, with (109), (110), and repeated applications of Lemma 7.8, we get

$$\emptyset; \emptyset \vdash \underbrace{[v_0/\mathtt{this}, \overline{v/x}][\overline{V/X}]\sigma_1 e''}_{=e'} : [\overline{V/X}]\sigma_1 U'$$

To finish this case, we still need to show that $\emptyset \vdash [\overline{V/X}]\sigma_1 U' \leq T$. (The claim then follows with rule EXP-SUBSUME.) From (97) we get with (123) and Lemma 7.1 that

$$\emptyset \vdash [\overline{V/X}]\sigma_1 U' \leq [\overline{V/X}]\sigma_1\sigma_3 W$$

Moreover, with (122) and transitivity of subtyping we then get

$$\emptyset \vdash [\overline{V/X}]\sigma_1 U' \leq [\overline{V/X}]\sigma_2 W$$

Ultimately, we have

$$[\overline{V/X}]\sigma_2 W \overset{(90)}{=} [\overline{V/X}]U \overset{(84)}{=} T$$

– *Case* other rules: Impossible.

End case distinction on the rule used to reduce e.

- *Case* rule EXP-INVOKE-S: Then

$$\frac{\begin{array}{c}\mathsf{smtype}_{\emptyset}(m, I\langle\overline{W}\rangle[\overline{T}]) = \langle\overline{X}\rangle\,\overline{U\,x} \to U \;\texttt{where}\;\overline{\mathcal{P}} \qquad (\forall i)\ \emptyset;\emptyset \vdash e_i : [\overline{V/X}]U_i \\ \emptyset \Vdash [\overline{V/X}]\overline{\mathcal{P}} \qquad 1 \notin \mathsf{pos}^+(I) \text{ or } (\exists i)\ \emptyset \Vdash T_i\,\mathsf{mono} \qquad \emptyset \vdash \overline{T}, \overline{V}\ \mathsf{ok}\end{array}}{\emptyset;\emptyset \vdash \underbrace{I\langle\overline{W}\rangle[\overline{T}].m\langle\overline{V}\rangle(\overline{e})}_{=e} : \underbrace{[\overline{V/X}]U}_{=T}}\ \textsc{exp-invoke-s} \tag{124}$$ {eq:invoke-s::theorem:preservation-stepsto}

Expanding the definition of smtype yields:

$$\frac{\begin{array}{c}\texttt{interface}\ I\langle\overline{Y'}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{Q'}\,\{\,\overline{m : \texttt{static}\ msig}\ \ldots\} \\ \emptyset \Vdash \overline{T}\ \texttt{implements}\ I\langle\overline{W}\rangle \qquad m = m_k\end{array}}{\mathsf{smtype}_{\emptyset}(m, I\langle\overline{W}\rangle[\overline{T}]) = \underbrace{[\overline{W/Y'}, \overline{T/Y}]\,msig_k}_{=\langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}}}\ \textsc{mtype-static} \tag{125}$$ {eq:smtype::theorem:preservation-stepsto}

Define $\sigma_2 = [\overline{W/Y'}, \overline{T/Y}]$ and assume

$$msig_k = \langle\overline{X''}\rangle\,\overline{U''\,x''} \to U''\ \texttt{where}\ \overline{P}$$

Then

$$\overline{X''} = \overline{X} \tag{126}$$ {eq:X''=X::theorem:preservation-stepsto}

$$\overline{x''} = \overline{x} \tag{127}$$ {eq:x''=x::theorem:preservation-stepsto}

$$\sigma_2(\overline{U''}, U'', \overline{P}) = (\overline{U}, U, \overline{\mathcal{P}}) \tag{128}$$ {eq:subst2-eq::theorem:preservation-stepsto}

By looking at the form of e, we see that $e \longmapsto e'$ must have been performed by rule DYN-INVOKE-S. Thus,

$$\frac{\mathsf{getsmdef}(m, I\langle\overline{W}\rangle, \overline{T}) = \langle\overline{X'}\rangle\,\overline{U'\,x'} \to U'\ \texttt{where}\ \overline{\mathcal{P}'}\,\{e''\}}{I\langle\overline{W}\rangle[\overline{T}].m\langle\overline{V}\rangle(\overline{v}) \longmapsto \underbrace{[\overline{v/x}][\overline{V/X}]e''}_{=e'}}\ \textsc{dyn-invoke-s} \tag{129}$$ {eq:dyn-invoke-s::theorem:preservation-stepsto}

$$\overline{v} = \overline{e} \tag{130}$$ {eq:v=e::theorem:preservation-stepsto}

Expanding the definition of $\mathsf{getsmdef}$ yields together with criterion WF-IFACE-1 that

$$\frac{\begin{array}{c}\texttt{interface}\ I\langle\overline{Y'}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{Q'}\,\{\,\overline{m : \texttt{static}\ msig}\ \ldots\} \qquad m = m_k \\ (\sigma_1, \texttt{implementation}\langle\overline{Z}\rangle\ I\langle\overline{W'}\rangle\ [\,\overline{N'}^{l}\,]\ \texttt{where}\ \overline{Q}\ \{\,\overline{\texttt{static}\ mdef}\ldots\}) = \mathsf{minimpl}\,\mathscr{M}\end{array}}{\mathsf{getsmdef}(m, I\langle\overline{W}\rangle, \overline{T}^{l}) = \underbrace{\sigma_1\,mdef_k}_{=\langle\overline{X'}\rangle\,\overline{U'\,x'} \to U'\ \texttt{where}\ \overline{\mathcal{P}'}\,\{e''\}}}\ \textsc{dyn-mdef-s} \tag{131}$$ {eq:getsmdef::theorem:preservation-stepsto}

where

$$\begin{array}{l}\mathscr{M} = \{(\sigma, \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{N}^{l}\,]\ \ldots) \\ \qquad \mid \mathsf{dom}(\sigma) = \overline{X}, (\forall i \in [l])\ N_i = \texttt{Object} \text{ or } T_i \trianglelefteq_c \sigma N_i\}\end{array}$$

Assume

$$mdef_k = \langle\overline{X'}\rangle\, \overline{U'''\, x'} \to U''' \texttt{ where } \overline{P'}\, \{e'''\}$$

Then

$$\sigma_1(\overline{U'''}, U''', \overline{P'}, e''') = \overline{U'}, U', \overline{P'}, e'' \tag{132}$$ {eq:subst1-eq::theorem:preservation-stepsto}

Because the underlying program is well-typed, we have by inverting rule OK-IMPL and criterion WF-IFACE-1

$$\overline{Q}, \overline{Z}; \emptyset \vdash mdef_k \ \mathsf{implements}\ \underbrace{[\overline{W'/Y'}, \overline{Y/N'}]}_{=\sigma_3}\, msig_k$$

We then have

$$\underbrace{\overline{Q}, \overline{Z}, \overline{P'}, \overline{X'}}_{=\Delta} \vdash \overline{U'''}, U''', \overline{P'}\ \mathsf{ok} \tag{133}$$ {eq:def-tenv::theorem:preservation-stepsto}

$$\Delta; \underbrace{\overline{x' : U'''}}_{=\Gamma} \vdash e''' : U''' \tag{134}$$ {eq:type-e'''::theorem:preservation-stepsto}

$$\overline{X'} = \overline{X''} \tag{135}$$ {eq:X'=X''2::theorem:preservation-stepsto}

$$\overline{U'''} = \sigma_3 \overline{U''} \tag{136}$$ {eq:U'''=U''::theorem:preservation-stepsto}

$$\overline{x'} = \overline{x''} \tag{137}$$ {eq:x'=x''2::theorem:preservation-stepsto}

$$\overline{P'} = \sigma_3 \overline{P} \tag{138}$$ {eq:P'=P::theorem:preservation-stepsto}

$$\Delta \vdash U''' \leq \sigma_3 U'' \tag{139}$$ {eq:U'''-st-U''::theorem:preservation-stepsto}

From (124) and (125) we have $\emptyset \Vdash \overline{T}\ \texttt{implements}\ I\langle\overline{W}\rangle$ and either $1 \notin \mathsf{pos}^+(I)$ or there exists some j with $\emptyset \Vdash T_j\ \texttt{mono}$. In the latter case, we have $T_j = N$ for some N by Lemma 7.12. If now $1 \notin \mathsf{pos}^+(I)$ or $j \notin \mathsf{pos}^+(I)$ then we immediately see with Lemma 7.25 that

$$impl = \texttt{implementation}\langle\overline{Z'}\rangle\ I\langle\overline{W''}\rangle\ [\overline{N''}]\ \texttt{where}\ \overline{Q''}\ \ldots$$

$$\mathsf{dom}(\sigma_4) = \overline{Z'}$$

$$\emptyset \Vdash \sigma_4 \overline{Q''} \tag{140}$$ {eq:entails-Q''::theorem:preservation-stepsto}

$$\overline{W} = \sigma_4 \overline{W''} \tag{141}$$ {eq:w=w''::theorem:preservation-stepsto}

$$(\forall i)\ \emptyset \vdash T_i \leq \sigma_4 N''_i \text{ with } T_i \neq \sigma_4 N''_i \text{ implying } i \in \mathsf{pos}^-(I) \tag{142}$$ {eq:Ti-st-Ni''::theorem:preservation-stepsto}

On the other hand, assume $j \in \mathsf{pos}^+(I)$ and $T_j = N$. Using Lemma 7.24, it is straightforward to show all claims except that it is not obvious why $T_j \neq \sigma_4 N''_j$ should imply $j \in \mathsf{pos}^-(I)$. For the sake of contradiction, assume $T_j \neq \sigma_4 N''_j$ and $j \notin \mathsf{pos}^-(I)$. We also have $j \in \mathsf{pos}^+(I)$, so with Lemma 7.24(v), we get $T_j = K$ for some K. But this is a contradiction to $T_j = N$.

With Lemma 7.10 and by looking at the definition of $\mathscr{M}$, we see that

$$(\sigma_4, impl) \in \mathscr{M} \tag{143}$$ {eq:in-M::theorem:preservation-stepsto}

Thus, with (131) and the definition of $\mathsf{minimpl}$:

$$(\forall i)\ \sigma_1 N'_i \trianglelefteq_c \sigma_4 N''_i \tag{144}$$ {eq:N'-st-N''::theorem:preservation-stepsto}

With (140) and criterion WF-PROG-4 we get $\emptyset \Vdash \sigma_1 \overline{Q}$. With Lemma 7.1 then

$$\emptyset \Vdash [\overline{V/X}]\sigma_1 \overline{Q} \tag{145}$$ {eq:entails-Q::theorem:preservation-stepsto}

From (143), (131), (144), and criterion WF-PROG-2 we get $\sigma_4\overline{W''} = \sigma_1\overline{W'}$, so with (141)

$$\overline{W} = \sigma_1\overline{W'} \tag{146}$$ {eq:w=w'::theorem:preservation-stepsto}

We get from criterion WF-IFACE-4 that $\overline{Y} \cap \mathsf{ftv}(\overline{P}) = \emptyset$. W.l.o.g., $\mathsf{dom}(\sigma_1) = \overline{Z} \cap \mathsf{ftv}(\overline{P}) = \emptyset$. Hence,

$$\sigma_2\overline{P} = [\overline{W/Y'}]\overline{P} \overset{(146)}{=} [\overline{\sigma_1 W'/Y'}]\overline{P} = \sigma_1[\overline{W'/Y'}]\overline{P} = \sigma_1\sigma_3\overline{P}$$

From (124) we have $\emptyset \Vdash [\overline{V/X}]\overline{\mathcal{P}}$ and from (125) we have $[\overline{V/X}]\overline{\mathcal{P}} = [\overline{V/X}]\sigma_2\overline{P}$. Thus, $\emptyset \Vdash [\overline{V/X}]\sigma_1\sigma_3\overline{P}$, so with (138) $\emptyset \Vdash [\overline{V/X}]\sigma_1\overline{P'}$. With (145) and (133) then

$$\emptyset \Vdash [\overline{V/X}]\sigma_1\Delta \tag{147}$$ {eq:entails-tenv2::theorem:preservation-stepsto}

Next, we show that $(\forall i)\ \emptyset; \emptyset \vdash v_i : [\overline{V/X}]\sigma_1 U_i'''$. Assume some i. W.l.o.g., $\mathsf{dom}(\sigma_1) = \overline{Z} \cap \mathsf{ftv}(U_i'') = \emptyset$.

Case distinction on whether or not $\overline{Y} \cap \mathsf{ftv}(U_i'') = \emptyset$.

– *Case* $\overline{Y} \cap \mathsf{ftv}(U_i'') = \emptyset$: Then

$$U_i \overset{(128)}{=} \sigma_2 U_i'' = [\overline{W/Y'}]U_i'' \overset{(146)}{=} [\overline{\sigma_1 W'/Y'}]U_i'' = \sigma_1[\overline{W'/Y'}]U_i'' = \sigma_1\sigma_3 U_i'' \overset{(136)}{=} \sigma_1 U_i'''$$

Using reflexivity of subtyping, we get

$$\emptyset \vdash U_i \leq \sigma_1 U_i'''$$

– *Case* $\overline{Y} \cap \mathsf{ftv}(U_i'') \neq \emptyset$: By criterion WF-IFACE-4 we than have $U_i'' = Y_j$ for some $j \in [l]$. Then

$$U_i \overset{(128)}{=} \sigma_2 U_i'' = \sigma_2 Y_j = T_j$$

We also have

$$\sigma_1 N_j' \overset{\text{def. of } \sigma_3}{=} \sigma_1\sigma_3 Y_j = \sigma_1\sigma_3 U_i'' \overset{(136)}{=} \sigma_1 U_i'''$$

By definition of $\mathscr{M}$ we have that either $\sigma_1 N_j' = \texttt{Object}$ or $T_j \trianglelefteq_c \sigma_1 N_j'$. In both cases, using Lemma 7.18 if $T_j \trianglelefteq_c \sigma_1 N_j'$, we get

$$\emptyset \vdash U_i \leq \sigma_1 U_i'''$$

End case distinction on whether or not $\overline{Y} \cap \mathsf{ftv}(U_i'') = \emptyset$.

In both cases, we have established that $\emptyset \vdash U_i \leq \sigma_1 U_i'''$. With Lemma 7.1 we get $\emptyset \vdash [\overline{V/X}]U_i \leq [\overline{V/X}]\sigma_1 U_i'''$. From (124) and (130) we have $(\forall i)\ \emptyset; \emptyset \vdash v_i : [\overline{V/X}]U_i$, so we get with rule EXP-SUBSUME that

$$(\forall i)\ \emptyset; \emptyset \vdash v_i : [\overline{V/X}]\sigma_1 U_i''' \tag{148}$$ {eq:type-vi::theorem:preservation-stepsto}

Our next goal is to show that $\emptyset \vdash [\overline{V/X}]\sigma_1 U''' \leq [\overline{V/X}]U$. W.l.o.g., $\mathsf{dom}(\sigma_1) = \overline{Z} \cap \mathsf{ftv}(U'') = \emptyset$.

Case distinction on whether or not $\overline{Y} \cap \mathsf{ftv}(U'') = \emptyset$.

– *Case* $\overline{Y} \cap \mathsf{ftv}(U'') = \emptyset$: Then

$$U \overset{(128)}{=} \sigma_2 U'' = [\overline{W/Y'}]U'' \overset{(146)}{=} [\overline{\sigma_1 W'/Y'}]U'' = \sigma_1[\overline{W'/Y'}]U'' = \sigma_1\sigma_3 U''$$

Hence,

$$\emptyset \vdash \sigma_1\sigma_3 U'' \leq U$$

- *Case* $\overline{Y} \cap \mathsf{ftv}(U'') \neq \emptyset$: By criterion WF-IFACE-4 we than have $U'' = Y_j$ for some $j \in [l]$. Moreover, $j \notin \mathsf{pos}^-(I)$. Then

  $$\sigma_1\sigma_3 U'' = \sigma_1\sigma_3 Y_j \overset{\text{def. of } \sigma_3}{=} \sigma_1 N'_j \overset{(143),\text{def. of } \mathsf{minimpl}}{\trianglelefteq_c} \sigma_4 N''_j \overset{(142)}{=} T_i = \sigma_2 Y_j = \sigma_2 U'' \overset{(128)}{=} U$$

  We then get with Lemma 7.18

  $$\emptyset \vdash \sigma_1\sigma_3 U'' \leq U$$

End case distinction on whether or not $\overline{Y} \cap \mathsf{ftv}(U'') = \emptyset$.

In both cases, we have shown $\emptyset \vdash \sigma_1\sigma_3 U'' \leq U$ so with Lemma 7.1

$$\emptyset \vdash [\overline{V/X}]\sigma_1\sigma_3 U'' \leq [\overline{V/X}]U$$

From (139), (147), and Lemma 7.1 we have

$$\emptyset \vdash [\overline{V/X}]\sigma_1 U''' \leq [\overline{V/X}]\sigma_1\sigma_3 U''$$

With transitivity of subtyping, we then get

$$\emptyset \vdash [\overline{V/X}]\sigma_1 U''' \leq [\overline{V/X}]U \tag{149}$$ {eq:U'''-st-U::theorem:preservation-stepsto}

Now we combine the various results. Assume $\sigma_1 = [\overline{V'/Z}]$ W.l.o.g., $\mathsf{ftv}(\overline{V'}) \cap \mathsf{ftv}(\overline{X}) = \emptyset$. Thus, with (126) and (135) we have $[\overline{V/X}]\sigma_1 = [\overline{V/X}, \overline{V'/Z}]$. With (133) then

$$\mathsf{dom}(\Delta) \setminus \mathsf{dom}([\overline{V/X}]\sigma_1) = \emptyset$$

From (124) we get $\emptyset \vdash \overline{T}, \overline{V}$ ok. With Lemma 7.40 and the definition of $\mathscr{M}$ we then get $\emptyset \vdash \sigma_1 X$ ok for all $X \in \mathsf{dom}(\sigma_1)$. Thus,

$$\emptyset \vdash [\overline{V/X}]\sigma_1 Z \text{ for all } Z \in \mathsf{dom}([\overline{V/X}]\sigma_1)$$

With (147), (134), and Lemma 7.7 we now get

$$\emptyset; [\overline{V/X}]\sigma_1\Gamma \vdash [\overline{V/X}]\sigma_1 e''' : [\overline{V/X}]\sigma_1 U'''$$

With (148), the definition of Γ, and possibly repeated applications of Lemma 7.8 we then get

$$\emptyset; \emptyset \vdash [\overline{v/x}][\overline{V/X}]\sigma_1 e''' : [\overline{V/X}]\sigma_1 U'''$$

With (129) and (132) we get $[\overline{v/x}][\overline{V/X}]\sigma_1 e''' = e'$. Thus, with (124), (149), and rule EXP-SUBSUME we get

$$\emptyset; \emptyset \vdash e' : T$$

as required.

- *Case* rule EXP-NEW: Then $e = \mathtt{new}\, N(\overline{e})$. But this is a contradiction to $e \longmapsto e'$.
- *Case* rule EXP-CAST: Then

$$\frac{\emptyset \vdash N \text{ ok} \qquad \emptyset; \emptyset \vdash e_0 : T'}{\emptyset; \emptyset \vdash (N)\, e_0 : N} \text{ EXP-CAST}$$

with $e = (N)\, e_0$ and $T = N$. The reduction step $e \longmapsto e'$ must have been performed through rule DYN-CAST. Thus,

$$e' = e_0$$
$$e_0 = \mathtt{new}\, M(\overline{w})$$
$$M \trianglelefteq_c N$$

By Lemma 7.13 and a case analysis on the form of e_0, we know that

$$\emptyset; \emptyset \vdash e_0 : M$$

By Lemma 7.18 we have $\emptyset \vdash M \leq T$, so the claim $\emptyset; \emptyset \vdash e' : T$ follows with rule EXP-SUBSUME.

- *Case* rule EXP-SUBSUME: In this case, the claim follows directly from the I.H. and rule EXP-SUBSUME.

End case distinction on the last rule of the derivation of $\emptyset; \emptyset \vdash e : T$. □

Theorem 7.43 (Preservation)**.** *If $\emptyset; \emptyset \vdash e : T$ and $e \longrightarrow e'$ then $\emptyset; \emptyset \vdash e' : T$.*

PROOF. From $e \longrightarrow e'$ we get (by inverting rule DYN-CONTEXT) the existence of an evaluation context $\mathcal{E}$ and expressions e_0, e_0' such that $e = \mathcal{E}[e_0]$ and $e_0 \longmapsto e_0'$ and $\mathcal{E}[e_0'] = e'$. Hence, it suffices to show the following claim:

If $\emptyset; \emptyset \vdash \mathcal{E}[e] : T$ and $e \longmapsto e'$ then $\emptyset; \emptyset \vdash \mathcal{E}[e'] : T$.

The proof of this claim is by induction on $\mathcal{E}$. If $\mathcal{E} = []$, then the claim holds by Theorem 7.42. In all other cases, we first use Lemma 7.13 to obtain a derivation $\mathcal{D}$ for $\emptyset; \emptyset \vdash \mathcal{E}[e] : T'$ such that $\emptyset \vdash T' \leq T$ and $\mathcal{D}$ does not end with rule EXP-SUBSUME. Then the form of $\mathcal{E}$ uniquely determines the last rule $\mathfrak{r}$ used in $\mathcal{D}$. In each case, the claim then follows by the I.H. and applications of rules $\mathfrak{r}$ and EXP-SUBSUME. □

Definition 7.44 (Reflexive, transitive closure of evaluation relation)**.** *The relation $\longrightarrow^*$ denotes the reflexive, transitive closure of the evaluation relation $\longrightarrow$.*

Theorem 7.45 (Type soundness)**.** *If $\emptyset; \emptyset \vdash e : T$ then either*

(*i*) *e diverges, or*

(*ii*) *$e \longrightarrow^* v$ for some value v such that $\emptyset; \emptyset \vdash v : T$, or*

(*iii*) *$e \longrightarrow^* e'$ for some expression e such that e' is stuck on a bad cast.*

PROOF. Assume that the evaluation of e terminates. (Otherwise, e diverges and we are done.) Hence, there exists e' such that $e \longrightarrow^* e'$ and there exists no e'' such that $e' \longrightarrow e''$. With Theorem 7.43 we get (by induction on the length of the evaluation sequence) that $\emptyset; \emptyset \vdash e' : T$. Theorem 7.34 then gives us either that $e' = v$ for some value v or that e' is stuck on a bad cast. □

8 Determinacy of Evaluation

Lemma 8.1. *If $\mathsf{minimpl}\,\mathscr{M} = (\sigma_1, impl_1)$ and $\mathsf{minimpl}\,\mathscr{M} = (\sigma_2, impl_2)$ then $\sigma_1 = \sigma_2$ and $impl_1 = impl_2$.*

PROOF. Assume

$$impl_1 = \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{M}]\ \ldots$$
$$impl_2 = \texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\overline{N}]\ \ldots$$

Then $\mathsf{dom}(\sigma_1) = \overline{X}$, $\mathsf{dom}(\sigma_2) = \overline{Y}$, and, by definition of $\mathsf{minimpl}$, $\sigma_1\overline{M} \trianglelefteq_c \sigma_2\overline{N}$ and $\sigma_2\overline{N} \trianglelefteq_c \sigma_1\overline{M}$. (The notation $\overline{N} \trianglelefteq_c \overline{M}$ is short for $(\forall j)\ N_j \trianglelefteq_c M_j$.) The class graph is acyclic by criterion WF-PROG-5, so $\sigma_1\overline{M} = \sigma_2\overline{N}$. Criterion WF-PROG-6 then yields $impl_1 = impl_2$. Hence, $\overline{X} = \overline{Y}$ and $\overline{M} = \overline{N}$. We have $\overline{X} \subseteq \mathsf{ftv}(\overline{M})$ by criterion WF-IMPL-1, so with $\sigma_1\overline{M} = \sigma_2\overline{N}$ also $\sigma_1 = \sigma_2$. □

Lemma 8.2.

(i) *If* $\mathsf{getmdef}^{c}(m, N) = mdef$ *and* $\mathsf{getmdef}^{c}(m, N) = mdef'$ *then* $mdef = mdef'$.

(ii) *If* $\mathsf{getmdef}^{i}(m, N, \overline{N}) = mdef$ *and* $\mathsf{getmdef}^{i}(m, N, \overline{N}) = mdef'$ *then* $mdef = mdef'$.

(iii) *If* $\mathsf{getsmdef}(m, K, \overline{N}) = mdef$ *and* $\mathsf{getsmdef}(m, K, \overline{N}) = mdef'$ *then* $mdef = mdef'$.

PROOF.

(i) It is easy to see that both derivations must end with the same rule. The claim now follows with a routine rule induction.

(ii) We first prove that $N_1 \sqcup N_2 = M$ and $N_1 \sqcup N_2 = M'$ imply $M = M'$. This proof is by induction on the derivations of $N_1 \sqcup N_2 = M$ and $N_1 \sqcup N_2 = M'$. If both derivations end with the same rule then the claim follows directly (rules LUB-RIGHT and LUB-LEFT) or via the I.H. (rule LUB-SUPER). Otherwise, one derivation ends with rule LUB-RIGHT and the other with rule LUB-LEFT. Then $N_1 \trianglelefteq_c N_2$ and $N_2 \trianglelefteq_c N_1$, so $M = N_2 = N_1 = M'$ as the class graph is acyclic by criterion WF-PROG-5.

We then get that $\bigsqcup \mathscr{N} = M$ and $\bigsqcup \mathscr{N} = M'$ imply $M = M'$. From this we have that $\mathsf{contrib}_X(\overline{T}, \overline{N}) = M$ and $\mathsf{contrib}_X(\overline{T}, \overline{N}) = M'$ imply $M = M'$.

The claim now follows with Lemma 8.1.

(iii) Follows with Lemma 8.1.

□

Lemma 8.3. *If* $e \longmapsto e'$ *and* $e \longmapsto e''$ *then* $e' = e''$.

PROOF. *Case distinction* on the form of e.

- *Case* $e = x$: Impossible.
- *Case* $e = e_0.f$: Then both reductions are due to rule DYN-FIELD. Hence, $e_0 = \texttt{new}\, N(\overline{v})$, $\mathsf{fields}(N) = \overline{U\,f}$, $f = f_j$, and $e' = v_j$. Clearly, fields is deterministic. Moreover, field shadowing is not allowed, so f occurs exactly once in $\overline{f}$. Thus, $e'' = v_j = e'$.
- *Case* $e = e_0.m\langle\overline{T}\rangle(\overline{e})$: If $m = m^{c}$ then both reductions are due to rule DYN-INVOKE-C. Otherwise, $m = m^{i}$ and both reductions are due to rule DYN-INVOKE-I. In any case, the claim follows with Lemma 8.2.
- *Case* $e = K[\overline{T}].m\langle\overline{U}\rangle(\overline{e})$: The claim follows directly from Lemma 8.2.
- *Case* $e = \texttt{new}\, N(\overline{e})$: Impossible.
- *Case* $e = (N)\, e_0$: Obvious.

End case distinction on the form of e. □

Lemma 8.4. *Assume* $\mathcal{E}_1[e_1] = \mathcal{E}_2[e_2]$. *If* $e_1 \longmapsto e_1'$ *and* $e_2 \longmapsto e_2'$ *then* $\mathcal{E}_1 = \mathcal{E}_2$.

PROOF. We prove the claim by induction on the combined size of $\mathcal{E}_1$ and $\mathcal{E}_2$. A case distinction on the form of $\mathcal{E}_1[e_1]$ reveals that either $\mathcal{E}_1 = \square = \mathcal{E}_2$ or that $\mathcal{E}_1$ and $\mathcal{E}_2$ are identical up to subcontexts $\mathcal{E}_1'$ and $\mathcal{E}_2'$ with $\mathcal{E}_1'[e_1] = \mathcal{E}_2'[e_2]$. (This follows from the fact that, for all values v, $v \neq \mathcal{E}[\tilde{e}]$ such that $\tilde{e} \longmapsto \tilde{e}'$.) In the first case, the claim is immediate. In the second case, we get by the I.H. that $\mathcal{E}_1' = \mathcal{E}_2'$. But then also $\mathcal{E}_1 = \mathcal{E}_2$. □

Theorem 8.5 (Deterministic evaluation)**.** *If* $e \longrightarrow e'$ *and* $e \longrightarrow e''$ *then* $e' = e''$.

PROOF. By rule DYN-CONTEXT, we have that $e = \mathcal{E}[\tilde{e}]$, $\tilde{e} \longmapsto \tilde{e}'$, $e' = \mathcal{E}[\tilde{e}']$, and that $e = \mathcal{E}'[\hat{e}]$, $\hat{e} \longmapsto \hat{e}'$, $e'' = \mathcal{E}'[\hat{e}']$. By Lemma 8.4 we get $\mathcal{E} = \mathcal{E}'$, so we have $\tilde{e} = \hat{e}$. By Lemma 8.3 we then get $\tilde{e}' = \hat{e}'$. Hence, $e' = e''$. □

9 Equivalence of Quasi-algorithmic and Algorithmic Versions of Entailment and Subtyping

Definition 9.1 (Small derivations)**.** *A derivation $\mathcal{D}$ is* small *iff its direct subderivations are small and all its proper subderivations end with a conclusion other then the conclusion of $\mathcal{D}$.*

Lemma 9.2. *If $\mathcal{D}'$ is a subderivation of a small derivation $\mathcal{D}$, then $\mathcal{D}'$ is also small.*

Proof. By induction on the height of $\mathcal{D}$. If $\mathcal{D}' = \mathcal{D}$ then the claim is immediate. Otherwise, there exist a direct subderivation $\mathcal{D}''$ of $\mathcal{D}$ such that $\mathcal{D}'$ is a subderivation of $\mathcal{D}''$. By Definition 9.1, we know that $\mathcal{D}''$ is small. Applying the I.H. proves that $\mathcal{D}'$ is small. □

Notation 9.3.

- *We write $\mathcal{D} :: \mathcal{J}$ to denote that $\mathcal{D}$ is a derivation of judgment $\mathcal{J}$.*
- *We write $\mathcal{D}; \mathfrak{r} :: \mathcal{J}$ iff $\mathcal{D} :: \mathcal{J}$ and $\mathcal{D}$ ends with an application of rule $\mathfrak{r}$.*
- *We write $\mathsf{height}(\mathcal{D})$ to denote the height of a derivation $\mathcal{D}$.*

Lemma 9.4. *Let $\mathcal{J}$ be a judgment such that the inference rules defining $\mathcal{J}$ do not put restrictions on properties of derivations for judgments in the premises or the conclusion of a rule. For example, the following rule would not be allowed:*

$$\frac{\mathcal{D} :: \mathcal{J}' \qquad \mathsf{height}(\mathcal{D}) = 1}{\mathcal{J}}$$

Now suppose $\mathcal{D} :: \mathcal{J}$. Then there exists $\widehat{\mathcal{D}} :: \mathcal{J}$ such that $\widehat{\mathcal{D}}$ is small and $\mathsf{height}(\widehat{\mathcal{D}}) \leq \mathsf{height}(\mathcal{D})$.

Proof. By induction on the height of $\mathcal{D}$. If $\mathcal{D}$ is already small then we are done. Assume $\mathcal{D}$ is not small. Hence

$$\frac{\mathcal{D}_1 :: \mathcal{J}_1 \qquad \dots \qquad \mathcal{D}_n :: \mathcal{J}_n}{\mathcal{D} :: \mathcal{J}}\ \mathfrak{r}$$

By applying the I.H. we get $\mathcal{D}_i' :: \mathcal{J}_i$ for all $i \in [n]$ whereby $\mathcal{D}_i'$ is small and $\mathsf{height}(\mathcal{D}_i') \leq \mathsf{height}(\mathcal{D}_i)$. An application of rule $\mathfrak{r}$ now yields $\mathcal{D}' :: \mathcal{J}$ such that $\mathsf{height}(\mathcal{D}') \leq \mathsf{height}(\mathcal{D})$. If $\mathcal{D}'$ is small then we are done. Otherwise, we have the following situation:

$$\frac{\begin{array}{c}\mathcal{D}'' :: \mathcal{J} \\ \vdots\end{array}}{\mathcal{D} :: \mathcal{J}}\ \mathfrak{r}$$

with $\mathsf{height}(\mathcal{D}'') < \mathsf{height}(\mathcal{D})$. We now apply the I.H. to $\mathcal{D}'' :: \mathcal{J}$ and get $\mathcal{D}''' :: \mathcal{J}$ such that $\mathcal{D}'''$ is small and $\mathsf{height}(\mathcal{D}''') \leq \mathsf{height}(\mathcal{D}'') < \mathsf{height}(\mathcal{D})$. □

Definition 9.5 (Entailment goals)**.** *Let $\mathcal{D}$ be a derivation. The set of* entailment goals *occurring in $\mathcal{D}$ is defined as follows:*

$$\mathsf{goals}(\mathcal{D}) = \{R \mid \mathcal{D} \text{ contains a subderivation } \mathcal{D}'; \textsc{ent-q-alg-impl} :: \Delta \Vdash_{\mathrm{q}} R\}$$

Lemma 9.6. *If $\mathcal{D}'$ is a subderivation of $\mathcal{D}$ then $\mathsf{goals}(\mathcal{D}') \subseteq \mathsf{goals}(\mathcal{D})$.*

Proof. Obvious. □

Lemma 9.7. *Suppose $\mathcal{D}; \textsc{ent-q-alg-impl} :: \Delta \Vdash_{\mathrm{q}} R$. If $\mathcal{D}$ is small and $\mathcal{D}'$ is a proper subderivation of $\mathcal{D}$, then $R \notin \mathsf{goals}(\mathcal{D}')$.*

Proof. Assume $R \in \mathsf{goals}(\mathcal{D}')$. Hence, there exists a subderivation $\mathcal{D}''$; ENT-Q-ALG-IMPL $:: \Delta \Vdash_{\mathrm{q}} R$ of $\mathcal{D}'$. But this is a contradiction to $\mathcal{D}$ being small because $\mathcal{D}''$ is a proper subderivation of $\mathcal{D}$. $\square$

Lemma 9.8.

(i) *If $\mathcal{D}_1 :: \Delta \Vdash_{\mathrm{q}} \mathcal{P}$ and $\mathcal{D}_1$ is small, then $\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \mathcal{P}$ for all β and all $\mathscr{G}$ with $\mathsf{goals}(\mathcal{D}_1) \cap \mathscr{G} = \emptyset$.*

(ii) *If $\mathcal{D}_2 :: \Delta \Vdash_{\mathrm{q}}{}' \overline{U}\, \texttt{implements}\, I\langle\overline{V}\rangle$ and $\mathcal{D}_2$ is small and $\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{U}$, then $\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \overline{T}\, \texttt{implements}\, I\langle\overline{V}\rangle$ for all $\mathscr{G}$ with $\mathsf{goals}(\mathcal{D}_2) \cap \mathscr{G} = \emptyset$.*

(iii) *If $\mathcal{D}_3 :: \Delta \vdash_{\mathrm{q}} T \leq U$ and $\mathcal{D}_3$ is small, then $\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq U$ for all $\mathscr{G}$ with $\mathsf{goals}(\mathcal{D}_3) \cap \mathscr{G} = \emptyset$.*

Proof. We proceed by induction on the combined height of $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$.

(i) Suppose $\mathscr{G}$ is a set of entailment goals such that $\mathsf{goals}(\mathcal{D}_1) \cap \mathscr{G} = \emptyset$ and let $\beta \in \{\texttt{false}, \texttt{true}\}$.

Case distinction on the last rule used in $\mathcal{D}_1$.

- *Case* rule ENT-Q-ALG-EXTENDS: We then have $\mathcal{P} = T\, \texttt{extends}\, U$. By inverting the rule, we get $\mathcal{D}_1' :: \Delta \vdash_{\mathrm{q}} T \leq U$ such that $\mathcal{D}_1'$ is a subderivation of $\mathcal{D}_1$. From Lemma 9.2 we know that $\mathcal{D}_1'$ is small and Lemma 9.6 gives us $\mathsf{goals}(\mathcal{D}_1') \cap \mathscr{G} = \emptyset$. Applying part (iii) of the I.H. yields $\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq U$, so the claim follows with rule ENT-ALG-EXTENDS.
- *Case* rule ENT-Q-ALG-UP: We then have

$$\dfrac{(\forall i)\ \text{if } T_i \neq U_i \text{ then } i \in \mathsf{pos}^-(I) \qquad \begin{array}{c}(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' T_i \leq U_i \\ \mathcal{D}_1' :: \Delta \Vdash_{\mathrm{q}}{}' \overline{U}\, \texttt{implements}\, I\langle\overline{V}\rangle\end{array}}{\mathcal{D}_1 :: \Delta \Vdash_{\mathrm{q}} \underbrace{\overline{T}\, \texttt{implements}\, I\langle\overline{V}\rangle}_{=\mathcal{P}}}$$

Thus, we have

$$\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{U}$$

by rule ENT-ALG-LIFT. Moreover, $\mathcal{D}_1$ is small so

$$\mathcal{D}_1' \text{ is small}$$

by Lemma 9.2. Furthermore,

$$\mathsf{goals}(\mathcal{D}_1') \cap \mathscr{G} = \emptyset$$

with Lemma 9.6 and $\mathsf{goals}(\mathcal{D}_1) \cap \mathscr{G} = \emptyset$. Applying part (ii) of the I.H. now yields $\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \mathcal{P}$.

End case distinction on the last rule used in $\mathcal{D}_1$.

(ii) *Case distinction* on the last rule used in $\mathcal{D}_2$.

- *Case* rule ENT-Q-ALG-ENV: We then have $\overline{U}\, \texttt{implements}\, I\langle\overline{V}\rangle = \overline{G}\, \texttt{implements}\, I\langle\overline{V}\rangle$. Inverting the rule yields $R \in \Delta$ and $\overline{G}\, \texttt{implements}\, I\langle\overline{V}\rangle \in \mathsf{sup}(R)$. The claim now follows with the assumption $\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{G}$ by rule ENT-ALG-ENV.
- *Case* rule ENT-Q-ALG-IMPL: We have

$$\dfrac{\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots \qquad \Delta \Vdash_{\mathrm{q}} [\overline{W/X}]\overline{P}}{\mathcal{D}_2 :: \Delta \Vdash_{\mathrm{q}}{}' \underbrace{[\overline{W/X}](\overline{N}\, \texttt{implements}\, I\langle\overline{V'}\rangle)}_{=\overline{U}\, \texttt{implements}\, I\langle\overline{V}\rangle}} \tag{150}$$

{eq:init-deriv::lemma:completeness-alg-sub-entail-aux}

Suppose $\mathcal{D}'_i :: \Delta \Vdash_{\mathrm{q}} [\overline{W/X}]P_i$, let $\mathscr{G}$ be a set of entailment goals such that $\mathsf{goals}(\mathcal{D}_2) \cap \mathscr{G} = \emptyset$, and assume $\beta \in \{\mathtt{false}, \mathtt{true}\}$.
$\mathcal{D}_2$ is small by assumption, so

$$\mathcal{D}'_i \text{ is small}$$

with Lemma 9.2. Using Lemma 9.7 we get $\overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle \notin \mathsf{goals}(\mathcal{D}'_i)$. Moreover, $\mathsf{goals}(\mathcal{D}'_i) \subseteq \mathsf{goals}(\mathcal{D}_2)$. Because $\mathsf{goals}(\mathcal{D}_2) \cap \mathscr{G} = \emptyset$ we then have

$$\mathsf{goals}(\mathcal{D}'_i) \cap (\mathscr{G} \cup \{\overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle\}) = \emptyset$$

By part (i) of the I.H. we now get

$$\Delta; \mathscr{G} \cup \{\overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle\}; \mathtt{false} \vdash_{\mathrm{a}} [\overline{W/X}]P_i \tag{151}$$ {eq:entails-pi::lemma:completeness-alg-sub-entail-aux}

Moreover, $\overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle \in \mathsf{goals}(\mathcal{D}_2)$ by Definition 9.5 and $\mathsf{goals}(\mathcal{D}_2) \cap \mathscr{G} = \emptyset$ by the assumption, so

$$\overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle \notin \mathscr{G} \tag{152}$$ {eq:notin-goalset::lemma:completeness-alg-sub-entail-aux}

Furthermore, $\overline{U} = [\overline{W/X}]\overline{N}$ from (150) and $\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{U}$ by the assumption; hence

$$\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow [\overline{W/X}]\overline{N} \tag{153}$$ {eq:lift::lemma:completeness-alg-sub-entail-aux}

We conclude

$$\dfrac{\begin{array}{c}[\overline{W/X}]\overline{N}\ \mathtt{implements}\ I\langle\overline{V}\rangle \notin \mathscr{G} \quad \text{from (152) and (150)} \\ \mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\overline{N}]\ \mathtt{where}\ \overline{P}\ \ldots \quad \text{from (150)} \\ \Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow [\overline{W/X}]\overline{N} \quad \text{from (153)} \\ \overline{V} = [\overline{W/X}]\overline{V'} \quad \text{from (150)} \\ \Delta; \mathscr{G} \cup \{[\overline{W/X}]\overline{N}\ \mathtt{implements}\ I\langle\overline{V}\rangle\}; \mathtt{false} \Vdash_{\mathrm{a}} [\overline{W/X}]\overline{P} \quad \text{from (151)}\end{array}}{\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \overline{T}\ \mathtt{implements}\ I\langle\overline{V}\rangle}\ \textsc{ent-alg-impl}$$

- *Case* rule ENT-Q-ALG-IFACE: We then have $\overline{U} = J\langle\overline{W}\rangle$ such that

$$1 \in \mathsf{pos}^+(J) \tag{154}$$ {eq:pos-plus::lemma:completeness-alg-sub-entail-aux}

$$J\langle\overline{W}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle \tag{155}$$ {eq:extends-iface::lemma:completeness-alg-sub-entail-aux}

With Lemma 6.20

$$1 \in \mathsf{pos}^+(I) \tag{156}$$ {eq:pos-plus-I::lemma:completeness-alg-sub-entail-aux}

With the assumption $\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{U}$ we get $\overline{T} = T$ for some T and

$$\Delta \vdash_{\mathrm{a}}{}' T \leq J\langle\overline{W}\rangle$$

$$\beta \text{ or } T = J\langle\overline{W}\rangle \text{ or } 1 \in \mathsf{pos}^-(I) \tag{157}$$ {eq:3or::lemma:completeness-alg-sub-entail-aux}

With (155), rule ENT-Q-ALG-IFACE, and Lemma 6.5 we get

$$\Delta \vdash_{\mathrm{a}}{}' T \leq I\langle\overline{V}\rangle \tag{158}$$ {eq:t-sub-iv::lemma:completeness-alg-sub-entail-aux}

Case distinction on the form of T.

- *Case* $T \neq J\langle\overline{W}\rangle$: With (157) we get β or $1 \in \mathsf{pos}^-(I)$. With (158) and rule ENT-ALG-LIFT we get $\Delta; \beta; I \vdash_{\mathrm{a}} T \uparrow I\langle\overline{V}\rangle$. With (156) and rule ENT-ALG-IFACE$_1$ we get $\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \overline{T}\ \mathtt{implements}\ I\langle\overline{V}\rangle$.
- *Case* $T = J\langle\overline{W}\rangle$: The claim then follows with (154), (155), and rule ENT-ALG-IFACE$_2$.

End case distinction on the form of T.

End case distinction on the last rule used in $\mathcal{D}_2$.

(iii) *Case distinction* on the last rule used in $\mathcal{D}_3$.

- *Case* rule SUB-Q-ALG-KERNEL: By inverting the rule, we get $\Delta \vdash_{\mathrm{q}}{}' T \leq U$, so $\Delta \vdash_{\mathrm{a}}{}' T \leq U$ by SUB-ALG-KERNEL-QUASI, so $\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq U$ by SUB-ALG-KERNEL.
- *Case* rule SUB-Q-ALG-IMPL: We have $U = I\langle \overline{V} \rangle$ for some $I\langle \overline{V} \rangle$ such that

$$\frac{\Delta \vdash_{\mathrm{q}}{}' T \leq T' \qquad \mathcal{D}_3' :: \Delta \Vdash_{\mathrm{q}}{}' T' \,\texttt{implements}\, I\langle \overline{V} \rangle}{\mathcal{D}_3 :: \Delta \vdash_{\mathrm{q}} T \leq I\langle \overline{V} \rangle}$$

By SUB-ALG-KERNEL-QUASI we have $\Delta \vdash_{\mathrm{a}}{}' T \leq T'$, so

$$\Delta; \texttt{true}; I \vdash_{\mathrm{a}} T \uparrow T'$$

by rule ENT-ALG-LIFT. Because $\mathcal{D}_3$ is small, we get with Lemma 9.2 that

$$\mathcal{D}_3' \text{ is small}$$

Moreover, by Lemma 9.6 $\mathsf{goals}(\mathcal{D}_3') \subseteq \mathsf{goals}(\mathcal{D}_3)$, so with the assumption $\mathsf{goals}(\mathcal{D}_3) \cap \mathscr{G} = \emptyset$ we have

$$\mathsf{goals}(\mathcal{D}_3') \cap \mathscr{G} = \emptyset$$

Applying part (ii) of the I.H. now yields

$$\Delta; \mathscr{G}; \texttt{true} \Vdash_{\mathrm{a}} T \,\texttt{implements}\, I\langle \overline{V} \rangle$$

so we get $\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq I\langle \overline{V} \rangle$ by rule SUB-ALG-IMPL.

End case distinction on the last rule used in $\mathcal{D}_3$. □

Theorem 9.9 (Completeness of algorithmic entailment and subtyping)**.**

(i) If $\mathcal{D}_1 :: \Delta \Vdash_{\mathrm{q}} \mathcal{P}$ then $\Delta \Vdash_{\mathrm{a}} \mathcal{P}$

(ii) If $\mathcal{D}_2 :: \Delta \Vdash_{\mathrm{q}}{}' \mathcal{R}$ then $\Delta \Vdash_{\mathrm{a}} \mathcal{R}$.

(iii) If $\mathcal{D}_3 :: \Delta \vdash_{\mathrm{q}} T \leq U$ then $\Delta \vdash_{\mathrm{a}} T \leq U$.

(iv) If $\mathcal{D}_4 :: \Delta \vdash_{\mathrm{q}}{}' T \leq U$ then $\Delta \vdash_{\mathrm{a}}{}' T \leq U$.

PROOF. By Lemma 9.4 we may safely assume that $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$ are small.

(i) Follows from Lemma 9.8 and rule ENT-ALG-MAIN.

(ii) With Lemma 6.4 we have $\Delta; \texttt{false}; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{T}$ for all I and all $\overline{T}$. The claim now follows from Lemma 9.8 and rule ENT-ALG-MAIN.

(iii) Follows from Lemma 9.8 and rule SUB-ALG-MAIN.

(iv) Follows directly with rule SUB-ALG-KERNEL-QUASI. □

Definition 9.10 ($\Delta \vdash_{\mathrm{q}} \overline{T} \uparrow \overline{T}$)**.**

ENT-Q-ALG-LIFT

$$\frac{(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' T_i \leq U_i \qquad \beta \text{ or } \big((\forall i) \text{ if } T_i \neq U_i \text{ then } i \in \mathsf{pos}^-(I)\big)}{\Delta; \beta; I \vdash_{\mathrm{q}} \overline{T} \uparrow \overline{U}}$$

Lemma 9.11. *If* $\Delta \Vdash_{\mathrm{q}}{}' \overline{U}\, \texttt{implements}\, I\langle\overline{V}\rangle$ *and* $\Delta; \texttt{false}; I \vdash_{\mathrm{q}} \overline{T} \uparrow \overline{U}$ *then* $\Delta \Vdash_{\mathrm{q}} \overline{T}\, \texttt{implements}\, I\langle\overline{V}\rangle$.

Proof. From the assumption $\Delta; \texttt{false}; I \vdash_{\mathrm{q}} \overline{T} \uparrow \overline{U}$ we get

$$(\forall i)\ \Delta \vdash_{\mathrm{q}}{}' T_i \leq U_i$$
$$(\forall i)\ \text{if } T_i \neq U_i \text{ then } i \in \mathsf{pos}^-(I)$$

The claim now follows with rule ent-q-alg-up. □

Lemma 9.12.

(*i*) *If* $\Delta \vdash_{\mathrm{a}}{}' T \leq U$ *then* $\Delta \vdash_{\mathrm{q}}{}' T \leq U$.

(*ii*) *If* $\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{U}$ *then* $\Delta; \beta; I \vdash_{\mathrm{q}} \overline{T} \uparrow \overline{U}$

Proof. Obvious □

Lemma 9.13.

(*i*) *If* $\mathcal{D}_1 :: \Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \overline{T}\, \texttt{implements}\, I\langle\overline{V}\rangle$ *then* $\Delta \Vdash_{\mathrm{q}}{}' \overline{U}\, \texttt{implements}\, I\langle\overline{V}\rangle$ *for some* $\overline{U}$ *with* $\Delta; \beta; I \vdash_{\mathrm{q}} \overline{T} \uparrow \overline{U}$.

(*ii*) *If* $\mathcal{D}_2 :: \Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq U$ *then* $\Delta \vdash_{\mathrm{q}} T \leq U$.

Proof. We proceed by induction on the combined height of $\mathcal{D}_1$ and $\mathcal{D}_2$.

(i) *Case distinction* on the last rule used in $\mathcal{D}_1$.

- *Case* ent-alg-extends: Impossible.
- *Case* ent-alg-env: Inverting the rule yields

$$R \in \Delta$$
$$\overline{G}\, \texttt{implements}\, I\langle\overline{V}\rangle \in \mathsf{sup}(R)$$
$$\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{G}$$

  By Lemma 9.12 we get $\Delta; \beta; I \vdash_{\mathrm{q}} \overline{T} \uparrow \overline{G}$. With rule ent-q-alg-env we have $\Delta \Vdash_{\mathrm{q}}{}' \overline{G}\, \texttt{implements}\, I\langle\overline{V}\rangle$. Defining $\overline{U} = \overline{G}$ finishes this case.

- *Case* ent-alg-impl: Inverting the rule yields

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots \tag{159}$$ {eq:impl-def::lemma:soundness-alg-sub-entail-aux}

$$\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow [\overline{W/X}]\overline{N} \tag{160}$$ {eq:lift::lemma:soundness-alg-sub-entail-aux}

$$\overline{V} = [\overline{W/X}]\overline{V'}$$

$$\Delta; \mathscr{G} \cup \{[\overline{W/X}]\overline{N}\, \texttt{implements}\, I\langle\overline{V}\rangle\}; \texttt{false} \Vdash_{\mathrm{a}} [\overline{W/X}]\overline{P} \tag{161}$$ {eq:entails-mc::lemma:soundness-alg-sub-entail-aux}

  Case distinction on the form of $[\overline{W/X}]P_i$.

  – *Case* $[\overline{W/X}]P_i = \overline{T'}\, \texttt{implements}\, J\langle\overline{U'}\rangle$: Assume Applying part (i) of the I.H. to (161) gives us the existence of $\overline{T''}$ such that

$$\Delta \Vdash_{\mathrm{q}}{}' \overline{T''}\, \texttt{implements}\, J\langle\overline{U'}\rangle$$
$$\Delta; \texttt{false}; J \vdash_{\mathrm{q}} \overline{T'} \uparrow \overline{T''}$$

  With Lemma 9.11 we then have

$$\Delta \Vdash_{\mathrm{q}} \overline{T'}\, \texttt{implements}\, J\langle\overline{U'}\rangle$$

- *Case* $[\overline{W/X}]P_i = T' \,\texttt{extends}\, U'$: Inverting the derivation in (161) yields

$$\Delta; \mathscr{G} \cup \{[\overline{W/X}]\overline{N} \,\texttt{implements}\, I\langle\overline{V}\rangle\} \vdash_{\mathrm{a}} T' \leq U'$$

Applying part (ii) of the I.H. yields $\Delta \vdash_{\mathrm{q}} T' \leq U'$. Thus

$$\Delta \Vdash_{\mathrm{q}} T' \,\texttt{extends}\, U'$$

with rule ENT-Q-ALG-EXTENDS.

End case distinction on the form of $[\overline{W/X}]P_i$. Thus, we have

$$\Delta \Vdash_{\mathrm{q}} [\overline{W/X}]\overline{P} \tag{162}$$ {eq:q-entails-mc::lemma:soundness-alg-sub-entail-aux}

With (159), (162), and rule ENT-Q-ALG-IMPL we get

$$\Delta \Vdash_{\mathrm{q}}{}' \, [\overline{W/X}]\overline{N} \,\texttt{implements}\, I\langle\overline{V}\rangle$$

Define $\overline{U} = [\overline{W/X}]\overline{N}$; then (160) finishes this case.

- *Case* ENT-ALG-IFACE$_1$: We then have $\overline{T} = T$ for some T. Inverting the rule yields

$$\Delta; \beta; I \vdash_{\mathrm{a}} T \uparrow I\langle\overline{V}\rangle$$
$$1 \in \mathsf{pos}^{+}(I)$$

By rule ENT-Q-ALG-IFACE, we have $\Delta \Vdash_{\mathrm{q}}{}' \; I\langle\overline{V}\rangle \,\texttt{implements}\, I\langle\overline{V}\rangle$. Defining $\overline{U} = I\langle\overline{V}\rangle$ finishes this case.

- *Case* ENT-ALG-IFACE$_2$: Then $\overline{T} = J\langle\overline{W}\rangle$ for some $J\langle\overline{W}\rangle$. Inverting the rule yields

$$1 \in \mathsf{pos}^{+}(J)$$
$$J\langle\overline{W}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle$$

The claim now follows directly with rule ENT-ALG-IFACE$_2$.

End case distinction on the last rule used in $\mathcal{D}_1$.

(ii) *Case distinction* on the last rule used in $\mathcal{D}_2$.

- *Case* SUB-ALG-KERNEL: Inverting the rule yields $\Delta \vdash_{\mathrm{a}}{}' \; T \leq U$. With Lemma 9.11 $\Delta \vdash_{\mathrm{q}}{}' \; T \leq U$, so the claim follows with rule SUB-Q-ALG-KERNEL.
- *Case* SUB-ALG-IMPL: Then $U = I\langle\overline{V}\rangle$ for some $I\langle\overline{V}\rangle$. Inverting the rule yields

$$\Delta; \mathscr{G}; \texttt{true} \Vdash_{\mathrm{a}} T \,\texttt{implements}\, I\langle\overline{V}\rangle$$

Applying part (i) of the I.H. gives us the existence of T' such that

$$\Delta \Vdash_{\mathrm{q}}{}' \; T' \,\texttt{implements}\, I\langle\overline{V}\rangle$$
$$\Delta; \texttt{true}; I \vdash_{\mathrm{q}} T \uparrow T'$$

Thus, we have $\Delta \vdash_{\mathrm{q}}{}' \; T \leq T'$. Rule SUB-Q-ALG-IMPL now proves the claim.

End case distinction on the last rule used in $\mathcal{D}_2$. □

Theorem 9.14 (Soundness of algorithmic entailment and subtyping)**.**

(*i*) *If* $\Delta \Vdash_{\mathrm{a}} \mathcal{P}$ *then* $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$.

(*ii*) *If* $\Delta \vdash_{\mathrm{a}} T \leq U$ *then* $\Delta \vdash_{\mathrm{q}} T \leq U$

PROOF.

(i) The derivation of $\Delta \Vdash_{\mathrm{a}} \mathcal{P}$ ends with rule ENT-ALG-MAIN. Inverting the rule yields

$$\mathcal{D} :: \Delta; \emptyset; \texttt{false} \Vdash_{\mathrm{a}} \mathcal{P}$$

Case distinction on the form of $\mathcal{P}$.

- *Case* $\mathcal{P} = T\,\texttt{extends}\,U$: Then $\mathcal{D}$ ends with rule ENT-ALG-EXTENDS. Inverting the rule yields $\Delta; \emptyset \vdash_{\mathrm{a}} T \leq U$. By Lemma 9.13 we get $\Delta \vdash_{\mathrm{q}} T \leq U$, thus $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$ by rule ENT-Q-ALG-EXTENDS,
- *Case* $\mathcal{P} = \overline{T}\,\texttt{implements}\,I\langle\overline{V}\rangle$: Applying Lemma 9.13 to $\mathcal{D}$ yields the existence of $\overline{U}$ such that

$$\Delta \Vdash_{\mathrm{q}}{}' \overline{U}\,\texttt{implements}\,I\langle\overline{V}\rangle$$
$$\Delta; \texttt{false}; I \vdash_{\mathrm{q}} \overline{T} \uparrow \overline{U}$$

  We then get $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$ by Lemma 9.11.

End case distinction on the form of $\mathcal{P}$.

(ii) The derivation of $\Delta \vdash_{\mathrm{a}} T \leq U$ ends with rule SUB-ALG-MAIN. Inverting the rule yields $\Delta; \emptyset \vdash_{\mathrm{a}} T \leq U$. The claim now follows with Lemma 9.13. □

10 Soundness, Completeness, and Termination of Entailment and Subtyping Algorithms

Definition 10.1. *A substitution σ is more general than a substitution σ' iff there exists a substitution τ such that $\sigma' = \tau\sigma$. In this case, we write $\sigma \preceq \sigma'$. If $\sigma \preceq \sigma'$ and $\sigma' \preceq \sigma$, then we write $\sigma \sim \sigma'$.*

Lemma 10.2. *$\sigma \sim \sigma'$ iff there exists a renaming τ such that $\sigma = \tau\sigma'$.*

PROOF. See, for example, [1]. □

Definition 10.3 (Unification modulo subtyping)**.** *A* unification problem modulo subtyping *is a triple*

$$\mathbb{U} = (\Delta, \overline{X}, \{T_1 \leq^? U_1, \ldots T_n \leq^? U_n\}) \quad .$$

A solution *of $\mathbb{U}$ is a substitution σ with $\mathsf{dom}(\sigma) \subseteq \overline{X}$ such that $\Delta \vdash_{\mathrm{q}}{}' \sigma T_i \leq \sigma U_i$ for all $i \in [n]$. We write $\mathsf{sol}(\mathbb{U})$ for the* set of all solutions *of $\mathbb{U}$. A* most general solution *of $\mathbb{U}$ is a solution σ such that for any other solution σ' it holds that $\sigma \preceq \sigma'$. We say that $\mathbb{U}$ is* well-formed *iff $\mathsf{ftv}(\Delta) \cap \overline{X} = \emptyset$ and $T_i = Y$ (or $U_i = Y$) implies $Y \notin \overline{X}$ for any $i \in [n]$.*

Definition 10.4. *The* weight of a type T with respect to a type environment Δ, *written* $\mathsf{weight}_\Delta(T)$, *is defined as follows:*

$$\begin{aligned}
&\mathsf{weight}_\Delta(X) = 1 + \mathsf{max}(\{\mathsf{weight}_\Delta(T) \mid X\,\texttt{extends}\,T \in \Delta\})\\
&\mathsf{weight}_\Delta(N) = 1\\
&\mathsf{weight}_\Delta(K) = 1
\end{aligned}$$

(By convention, $\mathsf{max}(\emptyset) = 0$.) The definition of weight *is proper (i.e., terminates) because Δ is contractive by criterion* WF-TENV-2.

Theorem 10.5 (Termination of $\texttt{unify}_{\leq}$)**.** $\texttt{unify}_{\leq}(\mathbb{U})$ *terminates for any $\mathbb{U}$.*

Proof. We first show that the rewrite rules in Fig. 14 terminate. The depth of a type, written $\mathsf{depth}(T)$, is defined as follows:

$$\begin{aligned}\mathsf{depth}(C\langle\overline{T}\rangle) &= \text{depth of } C \text{ in the class hierarchy}\\ \mathsf{depth}(I\langle\overline{T}\rangle) &= 1 + \text{depth of } I \text{ in the inheritance hierarchy}\\ \mathsf{depth}(X) &= 0\end{aligned}$$

We then define the measure of a unification problem modulo subtyping $(\Delta, \overline{X}, \{T_1 \leq^? U_1, \ldots, T_n \leq^? U_n\})$ as

$$(\sum_{i=1}^{n} \mathsf{weight}_\Delta(T_i), \sum_{i=1}^{n} \mathsf{depth}(T_i)) \in \mathbb{N} \times \mathbb{N}$$

It is easy to see that each transformation rule from Fig. 14 decreases this measure with respect to the usual lexicographic ordering on $\mathbb{N} \times \mathbb{N}$.

Termination of $\texttt{unify}_{\leq}$ now follows because $\texttt{unify}_{=}$ terminates. □

In the following, we extend $\overset{\Delta}{\Longrightarrow}$ to unification problems module subtyping:

$$\frac{\{\overline{T_i \leq^? U_i}\} \overset{\Delta}{\Longrightarrow} \{\overline{T_i' \leq^? U_i'}\}}{(\Delta, \overline{X}, \{\overline{T_i \leq^? U_i}\}) \Longrightarrow (\Delta, \overline{X}, \{\overline{T_i' \leq^? U_i'}\})}$$

Lemma 10.6. *If $\mathbb{U}$ is well-formed and $\mathbb{U} \Longrightarrow \mathbb{U}'$ then $\mathbb{U}'$ is well-formed.*

Proof. Easy. □

Lemma 10.7. *If $\mathbb{U}$ is well-formed and $\mathbb{U} \Longrightarrow \mathbb{U}'$ then $\mathsf{sol}(\mathbb{U}) = \mathsf{sol}(\mathbb{U}')$.*

Proof. Easy, using Lemma 10.6. □

Theorem 10.8 (Soundness of $\texttt{unify}_{\leq}$)**.** *If $\texttt{unify}_{\leq}(\mathbb{U}) = \texttt{OK}(\sigma)$ and $\mathbb{U}$ is well-formed, then σ is an idempotent, most general solution of $\mathbb{U}$.*

Proof. Follows with Lemma 10.7. □

Theorem 10.9 (Completeness of $\texttt{unify}_{\leq}$)**.** *If a well-formed unification problem modulo subtyping $\mathbb{U}$ has a solution, then $\texttt{unify}_{\leq}(\mathbb{U}) \neq \texttt{FAIL}$.*

Proof. Follows with Lemma 10.7. □

Theorem 10.10 (Soundness and completeness of entailment and subtyping algorithms)**.**

- $\Delta \Vdash_{\mathrm{a}} \mathcal{P}$ *iff* $\texttt{entails}(\Delta, \mathcal{P})$ *returns* $\texttt{true}$.
- $\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \mathcal{P}$ *iff* $\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})$ *returns* $\texttt{true}$.
- $\Delta \vdash_{\mathrm{a}} T \leq U$ *iff* $\texttt{sub}(\Delta, T, U)$ *returns* $\texttt{true}$.
- $\Delta; \mathscr{G} \vdash_{\mathrm{a}} T \leq U$ *iff* $\texttt{subAux}(\Delta, \mathscr{G}, T, U)$ *returns* $\texttt{true}$.
- $\Delta \vdash_{\mathrm{a}}' T \leq U$ *iff* $\texttt{sub'}(\Delta, T, U)$ *returns* $\texttt{true}$.
- $\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{U}$ *iff* $\texttt{lift}(\Delta, \beta, I, \overline{T}, \overline{U})$ *returns* $\texttt{true}$.

Proof. Completeness ($\Leftarrow$) follows by straightforward induction on the combined sizes of the given derivations. Soundness ($\Rightarrow$) follows by induction on the depth of the recursion. In the proofs, we use Theorem 10.5, Theorem 10.8, and Theorem 10.9. □

Lemma 10.11 (Transitivity of cls)**.** *If* $\mathscr{T}_3 \subseteq \mathsf{cls}_\Delta(\mathscr{T}_2)$ *and* $\mathscr{T}_2 \subseteq \mathsf{cls}_\Delta(\mathscr{T}_1)$ *then* $\mathscr{T}_3 \subseteq \mathsf{cls}_\Delta(\mathscr{T}_1)$.

PROOF. It suffices to show that $T \in \mathsf{cls}_\Delta(\mathscr{T}_2)$ implies $T \in \mathsf{cls}_\Delta(\mathscr{T}_1)$ for all T. The proof is by induction on the derivation of $T \in \mathsf{cls}_\Delta(\mathscr{T}_2)$.
Case distinction on the last rule used in the derivation of $T \in \mathsf{cls}_\Delta(\mathscr{T}_2)$.

- *Case* CLS-ID: Then $T \in \mathscr{T}_2$, so $T \in \mathsf{cls}_\Delta(\mathscr{T}_1)$ because $\mathscr{T}_2 \subseteq \mathsf{cls}_\Delta(\mathscr{T}_1)$ by assumption.
- *Case* CLS-UP: Then
$$\frac{U \in \mathsf{cls}_\Delta(\mathscr{T}_2) \qquad \Delta \vdash_{\mathrm{a}}{}' U \leq T}{T \in \mathsf{cls}_\Delta(\mathscr{T}_2)}$$
Applying the I.H. yields $U \in \mathsf{cls}_\Delta(\mathscr{T}_1)$. The claim now follows with rule CLS-UP.
- *Case* CLS-DECOMP: Then
$$\frac{B\langle\overline{T}\rangle \in \mathsf{cls}_\Delta(\mathscr{T}_2)}{T_i \in \mathsf{cls}_\Delta(\mathscr{T}_2)}$$
with $T = T_i$. Applying the I.H. yields $B\langle\overline{T}\rangle \in \mathsf{cls}_\Delta(\mathscr{T}_1)$. The claim now follows with rule CLS-DECOMP.

End case distinction on the last rule used in the derivation of $T \in \mathsf{cls}_\Delta(\mathscr{T}_2)$. □

Definition 10.12 (Entailment candidates)**.** *The set of* entailment candidates *of a constraint* $\mathcal{P}$ *with respect to a type environment* Δ*, written* $\mathsf{cand}_\Delta(\mathcal{P})$*, is defined as the least set closed under the following rules:*

CAND-CLS
$$\frac{\overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T})}{\overline{U}\,\texttt{implements}\,K \in \mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\,K)}$$

CAND-IMPL$_1$
$$\frac{\begin{array}{c}\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots\\ \overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad \overline{U'} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad P_i = \overline{W}\,\texttt{implements}\,L\end{array}}{\overline{U}\,\texttt{implements}\,[\overline{U'/X}]L \in \mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\,K)}$$

CAND-IMPL$_2$
$$\frac{\begin{array}{c}\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots\\ U \in \mathsf{cls}_\Delta(\overline{T}) \qquad \overline{U'} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad P_i = W\,\texttt{extends}\,W'\end{array}}{U\,\texttt{extends}\,[\overline{U'/X}]W' \in \mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\,K)}$$

CAND-EXTENDS
$$\frac{\mathcal{P} \in \mathsf{cand}_\Delta(T\,\texttt{implements}\,K)}{\mathcal{P} \in \mathsf{cand}_\Delta(T\,\texttt{extends}\,K)}$$

Definition 10.13 (left)**.** *For a constraint* $\mathcal{P}$*, we define* $\mathsf{left}(\mathcal{P})$ *as follows:*
$$\begin{aligned}\mathsf{left}(\overline{T}\,\texttt{implements}\,K) &= \overline{T}\\ \mathsf{left}(T\,\texttt{extends}\,U) &= U\end{aligned}$$

Lemma 10.14. *If* $\mathcal{P} \in \mathsf{cand}_\Delta(\mathcal{Q})$ *then* $\mathsf{left}(\mathcal{P}) \subseteq \mathsf{cls}_\Delta(\mathsf{left}(\mathcal{Q}))$.

PROOF. *Case distinction* on the last rule used in the derivation of $\mathcal{P} \in \mathsf{cand}_\Delta(\mathcal{Q})$.

- *Case* CAND-CLS: Then $\mathcal{P} = \overline{U}\,\texttt{implements}\,K$ and $\mathcal{Q} = \overline{T}\,\texttt{implements}\,K$ with $\overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T})$ and the claim is immediate.
- *Case* CAND-IMPL$_1$: We then have
$$\frac{\begin{array}{c}\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots\\ \overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad \overline{U'} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad P_i = \overline{W}\,\texttt{implements}\,L\end{array}}{\underbrace{\overline{U}\,\texttt{implements}\,[\overline{U'/X}]L}_{=\mathcal{P}} \in \mathsf{cand}_\Delta(\underbrace{\overline{T}\,\texttt{implements}\,K}_{=\mathcal{Q}})}$$
and the claim is immediate.

- *Case* CAND-IMPL$_2$: The claim follows analogously to the preceding case.
- *Case* CAND-EXTENDS: Then $\mathcal{P} = T\,\texttt{extends}\,K$ and $\mathcal{P}' \in \mathsf{cls}_\Delta(T\,\texttt{implements}\,K)$. Because this derivation cannot end with rule CAND-EXTENDS, the claim follows with the same argumentation as in one of the three preceding cases.

End case distinction on the last rule used in the derivation of $\mathcal{P} \in \mathsf{cand}_\Delta(\mathcal{Q})$. □

Lemma 10.15. *If* $\mathcal{P} \in \mathsf{cand}_\Delta(\mathcal{Q})$ *then* $\mathsf{cand}_\Delta(\mathcal{P}) \subseteq \mathsf{cand}_\Delta(\mathcal{Q})$.

PROOF. We show that $\mathcal{P}' \in \mathsf{cand}_\Delta(\mathcal{P})$ implies $\mathcal{P}' \in \mathsf{cand}_\Delta(\mathcal{Q})$ for all $\mathcal{P}'$.
Case distinction on the last rule used in the derivation of $\mathcal{P}' \in \mathsf{cand}_\Delta(\mathcal{P})$.

- *Case* CAND-CLS: We then have

$$\mathcal{P}' = \overline{U}\,\texttt{implements}\,K$$
$$\mathcal{P} = \overline{T}\,\texttt{implements}\,K$$
$$\overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T})$$

  By Lemma 10.14 we have $\overline{T} \subseteq \mathsf{cls}_\Delta(\mathsf{left}(\mathcal{Q}))$, so with Lemma 10.11

$$\overline{U} \subseteq \mathsf{cls}_\Delta(\mathsf{left}(\mathcal{Q})) \tag{163}$$ {eq:subset-left::lemma:cand-implies-subset}

  Case distinction on the last rule in the derivation of $\mathcal{P} \in \mathsf{cls}_\Delta(\mathcal{Q})$.

  - *Case* CAND-CLS: Then $\mathcal{Q} = \overline{V}\,\texttt{implements}\,K$. With (163) we have $\overline{U} \subseteq \mathsf{cls}_\Delta(\overline{V})$, so $\mathcal{P}' \in \mathsf{cls}_\Delta(\mathcal{Q})$ by rule CAND-CLS.
  - *Case* CAND-IMPL$_1$: Then

$$\frac{\begin{array}{c}\texttt{implementation}\langle\overline{X}\rangle\; I\langle\overline{V'}\rangle\;[\,\overline{N}\,]\;\texttt{where}\;\overline{P}\;\ldots\\ \overline{T} \subseteq \mathsf{cls}_\Delta(\overline{V}) \qquad \overline{T'} \subseteq \mathsf{cls}_\Delta(\overline{V}) \qquad P_i = \overline{W}\,\texttt{implements}\,K'\end{array}}{\overline{T}\,\texttt{implements}\,\underbrace{[\overline{T'/X}]K'}_{=K} \in \mathsf{cand}_\Delta(\underbrace{\overline{V}\,\texttt{implements}\,L}_{=\mathcal{Q}})}$$

    With (163) we have $\overline{U} \subseteq \mathsf{cls}_\Delta(\overline{V})$, so $\mathcal{P}' \in \mathsf{cls}_\Delta(\mathcal{Q})$ by rule CAND-IMPL$_1$.
  - *Case* CAND-IMPL$_2$: Impossible because $\mathcal{P}$ is not an `extends`-constraint.
  - *Case* CAND-EXTENDS: Then $\mathcal{Q} = V\,\texttt{extends}\,L$ and $\mathcal{P} \in \mathsf{cls}_\Delta(V\,\texttt{implements}\,L)$. Because this derivation cannot end with rule CAND-EXTENDS, the claim follows with the same argumentation as in one of the three preceding cases.

  End case distinction on the last rule in the derivation of $\mathcal{P} \in \mathsf{cls}_\Delta(\mathcal{Q})$.

- *Case* CAND-IMPL$_1$: We then have

$$\frac{\begin{array}{c}\texttt{implementation}\langle\overline{X}\rangle\; I\langle\overline{V}\rangle\;[\,\overline{N}\,]\;\texttt{where}\;\overline{P}\;\ldots\\ \overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad \overline{U'} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad P_i = \overline{W}\,\texttt{implements}\,L\end{array}}{\underbrace{\overline{U}\,\texttt{implements}\,[\overline{U'/X}]L}_{=\mathcal{P}'} \in \mathsf{cand}_\Delta(\underbrace{\overline{T}\,\texttt{implements}\,K}_{=\mathcal{P}})}$$

  By Lemma 10.14 we have $\overline{T} \subseteq \mathsf{cls}_\Delta(\mathsf{left}(\mathcal{Q}))$, so with Lemma 10.11

$$\overline{U} \subseteq \mathsf{cls}_\Delta(\mathsf{left}(\mathcal{Q})) \tag{164}$$ {eq:subset-left1::lemma:cand-implies-subset}
$$\overline{U'} \subseteq \mathsf{cls}_\Delta(\mathsf{left}(\mathcal{Q})) \tag{165}$$ {eq:subset-left2::lemma:cand-implies-subset}

  If now $\mathcal{Q} = \overline{W'}\,\texttt{implements}\,L'$ for some $\overline{W'}$ and L', then the claim follows with rule CAND-IMPL$_1$. Otherwise, $\mathcal{Q} = W'\,\texttt{extends}\,W''$. Because $\mathcal{P} \in \mathsf{cls}_\Delta(\mathcal{Q})$, we must have that $W'' = L'$ for some L'. With rule CAND-IMPL$_1$, we have $\mathcal{P}' \in \mathsf{cls}_\Delta(W'\,\texttt{implements}\,L')$, so the claim follows with rule CAND-EXTENDS.

- *Case* CAND-IMPL$_2$: The claim follows analogously to the preceding case, replacing CAND-IMPL$_1$ with CAND-IMPL$_2$.
- *Case* CAND-EXTENDS: Then $\mathcal{P} = T\,\texttt{extends}\,K$ and $\mathcal{P}' \in \mathsf{cls}_\Delta(T\,\texttt{implements}\,K)$. Because this derivation cannot end with rule CAND-EXTENDS, the claim follows with the same argumentation as in one of the three preceding cases.

End case distinction on the last rule used in the derivation of $\mathcal{P}' \in \mathsf{cand}_\Delta(\mathcal{P})$. □

Lemma 10.16. *If* $\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots$ *and* $\overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T})$ *then* $[\overline{U/X}]P_i \in \mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\,K)$ *for all* i.

PROOF. *Case distinction* on the form of C_i.

- *Case* $P_i = \overline{T'}\,\texttt{implements}\,K'$ for some $\overline{T'}$ and K': By criterion WF-IMPL-3 we have $\overline{T'} \subseteq \overline{X}$. Hence, $[\overline{U/X}]\overline{T'} \subseteq \overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T}\,\texttt{implements}\,K)$. Thus

$$\frac{\begin{array}{c}\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots\\ [\overline{U/X}]\overline{T'} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad \overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T}) \qquad P_i = \overline{T'}\,\texttt{implements}\,K'\end{array}}{[\overline{U/X}]P_i \in \mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\,K)}\ \text{CAND-IMPL}_1$$

- *Case* $P_i = T'\,\texttt{extends}\,T''$: By criterion WF-IMPL-3 we have $T' \in \overline{X}$. The claim now follows analogously to the preceding case, replace rule CAND-IMPL$_1$ with CAND-IMPL$_2$.

End case distinction on the form of C_i. □

Definition 10.17 (Call tree)**.** *The* call tree *of* $\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})$ *is a root node labeled* $\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})$ *whose subtrees are the call trees of all the direct recursive calls of* $\texttt{entailsAux}$ *and* $\texttt{subAux}$. *The call tree of* $\texttt{subAux}(\Delta, \mathscr{G}, T, U)$ *is defined analogously.*

Definition 10.18 (cache)**.** *We let* $\mathsf{cache}(\mathfrak{n})$ *denote the set of goals cached at node* $\mathfrak{n}$*; that is,*

$$\mathsf{cache}(\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})) = \mathscr{G}$$
$$\mathsf{cache}(\texttt{subAux}(\Delta, \mathscr{G}, T, U)) = \mathscr{G}$$

Lemma 10.19. *If* $\mathfrak{n}$ *is a node in the call tree of* $\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})$ *(or* $\texttt{subAux}(\Delta, \mathscr{G}, T, U)$*) then* $\mathsf{cache}(\mathfrak{n}) \subseteq \mathscr{G} \cup \mathsf{cand}_\Delta(\mathcal{P})$ *(or* $\mathsf{cache}(\mathfrak{n}) \subseteq \mathscr{G} \cup \mathsf{cand}_\Delta(T\,\texttt{extends}\,U)$*).*

PROOF. We prove the following, stronger claim:

> Suppose $\mathfrak{n}$ is a node in the call tree of $\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})$ (or $\texttt{subAux}(\Delta, \mathscr{G}, T, U)$). Define $\mathscr{M} = \mathsf{cand}_\Delta(\mathcal{P})$ (or $\mathscr{M} = \mathsf{cand}_\Delta(T\,\texttt{extends}\,U)$). Then $\mathsf{cache}(\mathfrak{n}) \subseteq \mathscr{G} \cup \mathscr{M}$ and $\mathsf{cand}_\Delta(\mathfrak{n}) \subseteq \mathscr{M}$.

(The notation $\mathsf{cand}_\Delta(\mathfrak{n})$ is defined as $\mathsf{cand}_\Delta(\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})) = \mathsf{cand}_\Delta(\mathcal{P})$ and $\mathsf{cand}_\Delta(\texttt{subAux}(\Delta, \mathscr{G}, T, U)) = \mathsf{cand}_\Delta(T\,\texttt{extends}\,U)$.)

The proof is by induction on the depth of $\mathfrak{n}$. If $\mathfrak{n}$ is the root node, then the claim is immediate. Otherwise, $\mathfrak{n}$ is the child of some node $\mathfrak{n}'$. Assume that the claim already holds for $\mathfrak{n}'$; that is,

$$\mathsf{cache}(\mathfrak{n}') \subseteq \mathscr{G} \cup \mathscr{M} \qquad (166)$$ {eq:ih1::lemma:cache-subset-goalset+cand}
$$\mathsf{cand}_\Delta(\mathfrak{n}') \subseteq \mathscr{M} \qquad (167)$$ {eq:ih2::lemma:cache-subset-goalset+cand}

Case distinction on the form of $\mathfrak{n}'$.

- *Case* $\mathfrak{n}' = \texttt{entailsAux}(\Delta', \mathscr{G}', \beta', \mathcal{P}')$: It is obvious that the type environment Δ remains constant throughout the whole call tree; hence, we may safely assume that $\Delta' = \Delta$.

  Case distinction on the line number of the call site corresponding to $\mathfrak{n}$.

- *Case* Line 5: Then $\mathsf{cache}(\mathfrak{n}) = \mathsf{cache}(\mathfrak{n}')$ and $\mathsf{cand}_\Delta(\mathfrak{n}) = \mathsf{cand}_\Delta(\mathfrak{n}')$, so the claim is immediate.
- *Case* Line 22: We have

$$\mathcal{P}' = \overline{T}^m \,\texttt{implements}\, I\langle\overline{V}\rangle$$
$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}^n \ldots$$
$$\texttt{lift}(\Delta, \beta', I, \overline{T}, [\overline{U/X}]\overline{N})$$
$$\overline{V} = [\overline{U/X}]\overline{V'}$$
$$[\overline{U/X}]\overline{N}\,\texttt{implements}\, I\langle\overline{V}\rangle \notin \mathscr{G}'$$
$$\mathscr{G}_0 = \mathscr{G}' \cup \{[\overline{U/X}]\overline{N}\,\texttt{implements}\, I\langle\overline{V}\rangle\}$$

and

$$\mathfrak{n} = \texttt{entailsAux}(\Delta, \mathscr{G}_0, \texttt{false}, [\overline{U/X}]P_i)$$

for some $i \in [n]$.
From $\texttt{lift}(\Delta, \beta', I, \overline{T}, [\overline{U/X}]\overline{N})$ we get with Theorem 10.10 that $\Delta \vdash_{\mathrm{a}}{}' T_j \leq [\overline{U/X}]N_j$ for all $j \in [m]$, hence

$$[\overline{U/X}]N_j \in \mathsf{cls}_\Delta(\overline{T}) \qquad (168) \qquad \texttt{\{eq:in-cls::lemma:cache-subset-goalset+cand\}}$$

for all $j \in [m]$ by rule CLS-UP. With (168) and rule CAND-CLS we get

$$[\overline{U/X}]\overline{N}\,\texttt{implements}\, I\langle\overline{V}\rangle \in \mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\, I\langle\overline{V}\rangle)$$

By (167) we have $\mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\, I\langle\overline{V}\rangle) \subseteq \mathscr{M}$, so we get

$$[\overline{U/X}]\overline{N}\,\texttt{implements}\, I\langle\overline{V}\rangle \in \mathscr{M}$$

Hence

$$\begin{aligned}
\mathsf{cache}(\mathfrak{n}) &= \mathscr{G}' \cup \{[\overline{U/X}]\overline{N}\,\texttt{implements}\, I\langle\overline{V}\rangle\} \\
&= \mathsf{cache}(\mathfrak{n}') \cup \{[\overline{U/X}]\overline{N}\,\texttt{implements}\, I\langle\overline{V}\rangle\} \\
&\overset{(166)}{\subseteq} \mathscr{G} \cup \mathscr{M} \cup \{[\overline{U/X}]\overline{N}\,\texttt{implements}\, I\langle\overline{V}\rangle\} \\
&= \mathscr{G} \cup \mathscr{M}
\end{aligned}$$

We still need to show $\mathsf{cand}_\Delta(\mathfrak{n}) \subseteq \mathscr{M}$. By criterion WF-IMPL-1, we have $\overline{X} \subseteq \mathsf{ftv}(\overline{N})$, so for each X_k there exists some N_j such that $X_k \in \mathsf{ftv}(N_j)$. Thus, U_k is a subterm of $[\overline{U/X}]N_j$. With (168) and possibly repeated applications of rule CLS-DECOMP, we get $U_k \in \mathsf{cls}_\Delta(\overline{T})$. Thus

$$\overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T})$$

With Lemma 10.16

$$[\overline{U/X}]P_i \in \mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\, I\langle\overline{V}\rangle)$$

Lemma 10.15 now yields

$$\mathsf{cand}_\Delta([\overline{U/X}]P_i) \subseteq \mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\, I\langle\overline{V}\rangle)$$

From (167) we have $\mathsf{cand}_\Delta(\overline{T}\,\texttt{implements}\, I\langle\overline{V}\rangle) \subseteq \mathscr{M}$. Moreover, $\mathsf{cand}_\Delta(\mathfrak{n}) = \mathsf{cand}_\Delta([\overline{U/X}]P_i)$, so $\mathsf{cand}_\Delta([\overline{U/X}]P_i) \subseteq \mathscr{M}$.

End case distinction on the line number of the call site corresponding to $\mathfrak{n}$.

- *Case* $\mathfrak{n}' = \texttt{subAux}(\Delta', \mathscr{G}', T', U')$: Again, we may safely assume $\Delta = \Delta'$. The call site corresponding to $\mathfrak{n}$ must be in ine 32. We then have

$$\mathscr{G}' = \mathscr{G}$$
$$U' = K \text{ for some } K$$
$$\mathfrak{n} = \texttt{entailsAux}(\Delta, \mathscr{G}, \texttt{true}, T' \,\texttt{implements}\, K)$$

We get

$$\mathsf{cache}(\mathfrak{n}) = \mathscr{G} = \mathsf{cache}(\mathfrak{n}') \overset{(166)}{\subseteq} \mathscr{G} \cup \mathscr{M}$$

and

$$\begin{aligned}\mathsf{cand}_\Delta(\mathfrak{n}) = \mathsf{cand}_\Delta(T' \,\texttt{implements}\, K) &\overset{\text{by rule CAND-EXTENDS}}{=} \mathsf{cand}_\Delta(T' \,\texttt{extends}\, K)\\ &= \mathsf{cand}_\Delta(\mathfrak{n}') \overset{(167)}{\subseteq} \mathscr{M}\end{aligned}$$

End case distinction on the form of $\mathfrak{n}'$. □

Definition 10.20 (Size of types and constraints)**.** *The* size *of a type or constraint (*$\mathsf{size}(T) \in \mathbb{N}^+$, $\mathsf{size}(\mathcal{P}) \in \mathbb{N}^+$ *) is defined as follows:*

$$\begin{aligned}\mathsf{size}(X) &= 1\\ \mathsf{size}(C\langle\overline{T}\rangle) &= 1 + \mathsf{size}(\overline{T})\\ \mathsf{size}(I\langle\overline{T}\rangle) &= 1 + \mathsf{size}(\overline{T})\\ \mathsf{size}(\overline{T} \,\texttt{implements}\, K) &= 1 + \mathsf{size}(K) + \mathsf{size}(\overline{T})\\ \mathsf{size}(T \,\texttt{extends}\, U) &= 1 + \mathsf{size}(T) + \mathsf{size}(U)\end{aligned}$$

Thereby, the size of a sequence of types $\overline{T}$ *is defined as* $\mathsf{size}(\overline{T}) = \sum_i \mathsf{size}(T_i)$.

Lemma 10.21. *Suppose* $\mathsf{cls}_\Delta(\mathscr{T})$ *is finite for every finite* $\mathscr{T}$. *Then* $\mathsf{cand}_\Delta(\mathcal{P})$ *is finite for all* $\mathcal{P}$.

Proof. We show that for all $\mathcal{P}$ there exists a $\delta(\mathcal{P}) \in \mathbb{N}^+$ such that $\mathsf{size}(\mathcal{Q}) \leq \delta(\mathcal{P})$ for all $\mathcal{Q} \in \mathsf{cand}_\Delta(\mathcal{P})$. The original claim then follows immediately because the set of types of a certain height is finite.

Let $\rho \in \mathbb{N}^+$ be a bound on the size of the constraints in the set $\mathscr{P}$ where

$$\mathscr{P} = \{P_i \mid \texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}^n \ \ldots, i \in [n]\}$$

Let $\vartheta(\mathcal{P}) \in \mathbb{N}^+$ be a bound on the size of the types in $\mathsf{cls}_\Delta(\mathsf{left}(\mathcal{P}))$. (Note that $\vartheta(\mathcal{P})$ exists because $\mathsf{cls}_\Delta(\mathsf{left}(\mathcal{P}))$ is finite by the assumption.) Define

$$\delta(\mathcal{P}) = \rho \cdot \vartheta(\mathcal{P}) \cdot \mathsf{size}(\mathcal{P})$$

Now suppose $\mathcal{Q} \in \mathsf{cand}_\Delta(\mathcal{P})$.
Case distinction on the last rule in the derivation of $\mathcal{Q} \in \mathsf{cand}_\Delta(\mathcal{P})$.

- *Case* cand-cls: Then $\mathcal{P} = \overline{Y} \,\texttt{implements}\, K$ and $\mathcal{Q} = \overline{U} \,\texttt{implements}\, K$ with $\overline{U} \subseteq \mathsf{cls}_\Delta(\overline{T})$. Hence, $\mathsf{size}(U_j) \leq \vartheta(\mathcal{P})$ for all j and the following inequality holds:

$$\begin{aligned}\mathsf{size}(\mathcal{Q}) &= 1 + \mathsf{size}(\overline{U}) + \mathsf{size}(K)\\ &\leq \vartheta(\mathcal{P}) + \mathsf{size}(\overline{T}) \cdot \vartheta(\mathcal{P}) + \mathsf{size}(K) \cdot \vartheta(K)\\ &= \vartheta(\mathcal{P}) \cdot \mathsf{size}(\mathcal{P})\\ &\leq \vartheta(\mathcal{P}) \cdot \mathsf{size}(\mathcal{P}) \cdot \rho = \delta(\mathcal{P})\end{aligned}$$

- *Case* CAND-IMPL$_1$: Then

$$\frac{\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots \qquad \overline{U}\subseteq\mathsf{cls}_\Delta(\overline{T}) \qquad \overline{U'}\subseteq\mathsf{cls}_\Delta(\overline{T}) \qquad P_i=\overline{W}\ \texttt{implements}\ L}{\underbrace{\overline{U}\ \texttt{implements}\ [\overline{U'/X}]L}_{=\mathcal{Q}} \in \mathsf{cand}_\Delta(\underbrace{\overline{T}\ \texttt{implements}\ K}_{=\mathcal{P}})}$$

  We have $\mathsf{size}(U_j)\leq\vartheta(\mathcal{P})$ and $\mathsf{size}(U'_k)\leq\vartheta(\mathcal{P})$ for all j,k. Moreover, $\mathsf{size}(P_i)\leq\rho$. Then the following inequality holds:

$$\begin{aligned}\mathsf{size}(\mathcal{Q}) &= 1+\mathsf{size}(\overline{U})+\mathsf{size}([\overline{U'/X}]L)\\ &\leq \vartheta(\mathcal{P})+\mathsf{size}(\overline{W})\cdot\vartheta(\mathcal{P})+\mathsf{size}(L)\cdot\vartheta(\mathcal{P})\\ &= \vartheta(\mathcal{P})\cdot\mathsf{size}(P_i)\\ &\leq \vartheta(\mathcal{P})\cdot\rho\cdot\mathsf{size}(\mathcal{P})=\delta(\mathcal{P})\end{aligned}$$

- *Case* CAND-IMPL$_2$: Analogously to the preceding case.
- *Case* CAND-EXTENDS: Then $\mathcal{P}=T\ \texttt{extends}\ K$ and $\mathcal{Q}\in\mathsf{cls}_\Delta(T\ \texttt{implements}\ K)$. Because this derivation cannot end with rule CAND-EXTENDS, the claim follows with the same argumentation as in one of the three preceding cases.

End case distinction on the last rule in the derivation of $\mathcal{Q}\in\mathsf{cand}_\Delta(\mathcal{P})$. □

Theorem 10.22. *The functions* $\texttt{entails}(\Delta,\mathcal{P})$, $\texttt{sub'}(\Delta,T,U)$, *and* $\texttt{sub}(\Delta,T,U)$ *terminate for all* $\mathcal{P}$, T, *and* U.

PROOF. By well-formedness criteria WF-TENV-1, WF-TENV-2, and WF-TENV-3, we know that Δ is finite and contractive and that $\mathsf{cls}_\Delta(\mathscr{T})$ is finite for every finite $\mathscr{T}$.

`sub'` terminates The weight function from Definition 10.4 is extended to recursive calls of `sub'` in the obvious way:

$$\mathsf{weight}(\texttt{sub'}(\Delta,T,U)=\mathsf{weight}_\Delta(T)+\mathsf{weight}_\Delta(U))$$

It is straightforward to verify that for each recursive call of `sub'`, the weight of the recursive call is strictly smaller than the weight of the original call. Moreover, the algorithms for checking class ($\trianglelefteq_c$) and interface ($\trianglelefteq_i$) inheritance terminate because the class and interface hierarchy is acyclic by criterion WF-PROG-5. Thus, `sub'` terminates.

`entails` terminates To prove that $\texttt{entails}(\Delta,\mathcal{P})$ terminates, we show that $\texttt{entailsAux}(\Delta,\mathscr{G},\beta,\mathcal{P})$ and $\texttt{subAux}(\Delta,\mathscr{G},T,U)$ terminate for finite $\mathscr{G}$. The claim then follows because $\texttt{entails}(\Delta,\mathcal{P})$ invokes `entailsAux` and `subAux` only with $\mathscr{G}=\emptyset$.

For the sake of a contradiction, assume that a concrete invocation of $\texttt{entailsAux}(\Delta,\mathscr{G},\beta,\mathcal{P})$ or $\texttt{subAux}(\Delta,\mathscr{G},T,U)$ diverges. It is easy to see that infinitely many calls of `entailsAux` or `subAux` must cause divergence:

- There are only finitely many choices for R in line 10 because Δ is finite.
- The algorithms for checking the relations $\mathcal{R}\in\mathsf{sup}(\mathcal{R})$, $i\in\mathsf{pos}^+(I)$, $i\in\mathsf{pos}^-(I)$ and $K\trianglelefteq_i K$ terminate because the interface graph is acyclic by criterion WF-PROG-5.
- The function `lift` terminates because `sub'` terminates as shown in the preceding case.
- The function $\texttt{unify}_\leq$ terminates by Theorem 10.5.

Hence, there exists a call tree $\mathfrak{t}$ of infinite size. We lead this to a contradiction by defining a measure μ from call tree nodes into $\mathbb{N} \times \mathbb{N}$ that strictly decreases (with respect to the usual lexicographic ordering on pairs) when moving from a node to any of its children.

Suppose the root node of $\mathfrak{t}$ is $\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})$ (or $\texttt{subAux}(\Delta, \mathscr{G}, T, U)$) and define $\mathscr{M} = \mathsf{cand}_\Delta(\mathcal{P})$ (or $\mathscr{M} = \mathsf{cand}_\Delta(T\,\texttt{extends}\,U)$). We have the assumption that $\mathsf{cls}_\Delta(\mathscr{T})$ is finite for every finite $\mathscr{T}$, so $\mathscr{M}$ is finite by Lemma 10.21. Because $\mathscr{G}$ is also finite, we now may define

$$\delta = |\mathscr{G}| + |\mathscr{M}| \in \mathbb{N}$$

We have by Lemma 10.19, $\mathsf{cache}(\mathfrak{n}) \subseteq \mathscr{G} \cup \mathscr{M}$ for all nodes $\mathfrak{n}$ in $\mathfrak{t}$. Hence, $(\delta - |\mathsf{cache}(\mathfrak{n})|, i) \in \mathbb{N} \times \mathbb{N}$ for all $i \in \mathbb{N}$ and all nodes $\mathfrak{n}$ in $\mathfrak{t}$. We now define the measure μ on nodes in $\mathfrak{t}$ as follows:

$$\begin{aligned}
\mu(\texttt{entailsAux}(\Delta', \mathscr{G}', \beta', \overline{T}\,\texttt{implements}\,K)) &= (\delta - |\mathscr{G}'|, 0) && \in \mathbb{N} \times \mathbb{N} \\
\mu(\texttt{entailsAux}(\Delta', \mathscr{G}', \beta', T\,\texttt{extends}\,K)) &= (\delta - |\mathscr{G}'|, 2) && \in \mathbb{N} \times \mathbb{N} \\
\mu(\texttt{subAux}(\Delta', \mathscr{G}', T, U)) &= (\delta - |\mathscr{G}'|, 1) && \in \mathbb{N} \times \mathbb{N}
\end{aligned}$$

Finally, we show that this measure strictly decreases when moving from a node to its children. Assume $\mathfrak{n}$ is a node in $\mathfrak{t}$ with children $\mathfrak{n}_1, \ldots, \mathfrak{n}_n$ and suppose $i \in [n]$.

Case distinction on the line number of the call site corresponding the $\mathfrak{n}_i$.

- *Case* Line 5: We have $\mathfrak{n} = \texttt{entailsAux}(\Delta, \mathscr{G}, \beta, T\,\texttt{extends}\,U)$, $n = 1$, and $\mathfrak{n}_1 = \texttt{subAux}(\Delta, \mathscr{G}, T, U)$. Hence,

  $$\mu(\mathfrak{n}_1) = (\delta - |\mathscr{G}|, 1) < (\delta - |\mathscr{G}|, 2) = \mu(\mathfrak{n})$$

- *Case* Line 22: We have

  $$\begin{aligned}
  \mathfrak{n} &= \texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \overline{T}\,\texttt{implements}\,I\langle\overline{V}\rangle) \\
  \mathscr{G}_0 &= \mathscr{G} \cup \{\sigma\overline{N}\,\texttt{implements}\,I\langle\overline{V}\rangle\} \\
  \sigma\overline{N}\,&\texttt{implements}\,I\langle\overline{V}\rangle \notin \mathscr{G} \\
  \mathfrak{n}_i &= \texttt{entailsAux}(\Delta, \mathscr{G}_0, \texttt{false}, \sigma P_i)
  \end{aligned}$$

  Thus, $|\mathscr{G}_0| = |\mathscr{G}| + 1$. Hence,

  $$\mu(\mathfrak{n}_i) = (\delta - |\mathscr{G}_0|, j) = (\delta - |\mathscr{G}| - 1, j) < (\delta - |\mathscr{G}|, 0) = \mu(\mathfrak{n})$$

  for some $j \in \{0, 1, 2\}$.

- *Case* Line 32: We have $\mathfrak{n} = \texttt{subAux}(\Delta, \mathscr{G}, T, K)$, $n = 1$, and $\mathfrak{n}_1 = \texttt{entailsAux}(\Delta, \mathscr{G}, \texttt{true}, T\,\texttt{implements}\,K)$. Thus

  $$\mu(\mathfrak{n}_1) = (\delta - |\mathscr{G}|, 0) < (\delta - |\mathscr{G}|, 1) = \mu(\mathfrak{n})$$

End case distinction on the line number of the call site corresponding the $\mathfrak{n}_i$.

`sub` terminates In the preceding case, we have shown that $\texttt{entailsAux}(\Delta, \mathscr{G}, \beta, \mathcal{P})$ and $\texttt{subAux}(\Delta, \mathscr{G}, T, U)$ terminate for finite $\mathscr{G}$. The claim follows immediately because $\texttt{sub}(\Delta, T, U)$ invokes `entailsAux` only with $\mathscr{G} = \emptyset$ and does not invoke `subAux` at all. □

11 Equivalence of Declarative and Algorithmic Versions of Expression Typing

Lemma 11.1.

(i) $\Delta \vdash T$ ok *if and only if* $\Delta \vdash_{\mathrm{a}} T$ ok.

(ii) $\Delta \vdash \mathcal{P}$ ok *if and only if* $\Delta \vdash_{\mathrm{a}} \mathcal{P}$ ok.

PROOF. Follows by straightforward induction on the combined size of the given derivations. Note that $\Delta \Vdash \mathcal{P}$ iff $\Delta \Vdash_{\mathrm{a}} \mathcal{P}$ by Theorem 6.36, Theorem 6.34, Theorem 9.14, and Theorem 9.9. $\square$

From now on, we use the equivalences and implications of the following corollary implicitly.

Corollary 11.2.

$\Delta \vdash T \leq U$	*iff*	$\Delta \vdash_{\mathrm{q}} T \leq U$	*(Theorem 6.34, Theorem 6.36)*
$\Delta \Vdash \mathcal{P}$	*iff*	$\Delta \Vdash_{\mathrm{q}} \mathcal{P}$	*(Theorem 6.34, Theorem 6.36)*
$\Delta \vdash_{\mathrm{q}} T \leq U$	*iff*	$\Delta \vdash_{\mathrm{a}} T \leq U$	*(Theorem 9.9, Theorem 9.14)*
$\Delta \vdash_{\mathrm{q}}' T \leq U$	*iff*	$\Delta \vdash_{\mathrm{a}}' T \leq U$	*(Rule* SUB-ALG-KERNEL-QUASI*)*
$\Delta \Vdash_{\mathrm{q}} \mathcal{P}$	*iff*	$\Delta \Vdash_{\mathrm{a}} \mathcal{P}$	*(Theorem 9.9, Theorem 9.14)*
$\Delta \vdash_{\mathrm{q}} T \leq G$	*implies*	$\Delta \vdash_{\mathrm{q}}' T \leq G$	*(Lemma 6.16)*
$\Delta \vdash_{\mathrm{q}}' T \leq U$	*implies*	$\Delta \vdash_{\mathrm{q}} T \leq U$	*(Rule* SUB-Q-ALG-KERNEL*)*
$\Delta \Vdash_{\mathrm{q}}' \mathcal{P}$	*implies*	$\Delta \Vdash_{\mathrm{q}} \mathcal{P}$	*(Lemma 6.19)*
$\Delta \vdash T$ ok	*iff*	$\Delta \vdash_{\mathrm{a}} T$ ok	*(Lemma 11.1)*
$\Delta \vdash \mathcal{P}$ ok	*iff*	$\Delta \vdash_{\mathrm{a}} \mathcal{P}$ ok	*(Lemma 11.1)*

Lemma 11.3 (Soundness of entailment for nillable constraints)**.** *If* $\Delta \Vdash^?_{\mathrm{a}} \overline{T^?}$ $\texttt{implements}\, I\langle\overline{W^?}\rangle \twoheadrightarrow \mathcal{R}$ *then* $\Delta \Vdash_{\mathrm{a}} \mathcal{R}$.

PROOF. We first show that

$$\Delta; \mathscr{G}; \beta \vdash^?_{\mathrm{a}} \overline{T^?}^n \uparrow \overline{U}^n \twoheadrightarrow \overline{V}^n \textit{ implies } \Delta; \mathscr{G}; \beta \vdash_{\mathrm{a}} \overline{V}^n \uparrow \overline{U}^n \tag{169}$$ {eq:aux-lemma::lemma:soundness-nil-entail}

From $\Delta; \mathscr{G}; \beta \vdash^?_{\mathrm{a}} \overline{T^?}^n \uparrow \overline{U}^n \twoheadrightarrow \overline{V}^n$ we get

$$(\forall i)\ T^?_i = \mathsf{nil} \text{ or } \Delta \vdash_{\mathrm{a}}' T^?_i \leq U_i$$
$$\beta \text{ or } \big((\forall i) \text{ if } T^?_i \neq U_i \text{ and } T^?_i \neq \mathsf{nil} \text{ then } i \in \mathsf{pos}^-(I)\big)$$
$$(\forall i)\ \text{ if } T^?_i = \mathsf{nil} \text{ then } V_i = U_i \text{ else } V_i = T^?_i$$

Hence, $(\forall i)$ $\Delta \vdash_{\mathrm{a}}' V_i \leq U_i$ and (β or (if $V_i \neq U_i$ then $i \in \mathsf{pos}^-(I)$)). We then have by rule ENT-ALG-LIFT that $\Delta; \mathscr{G}; \beta \vdash_{\mathrm{a}} \overline{V}^n \uparrow \overline{U}^n$.

We now prove that $\mathcal{D} :: \Delta; \mathscr{G}; \beta \Vdash^?_{\mathrm{a}} \overline{T^?}\ \texttt{implements}\, I\langle\overline{W^?}\rangle \twoheadrightarrow \mathcal{R}$ implies $\Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}} \mathcal{R}$ by induction on $\mathcal{D}$. The claim then follows with rule ENT-ALG-MAIN.
Case distinction on the last rule used in $\mathcal{D}$.

- *Case* Rule ENT-NIL-ALG-ENV: Then

$$R \in \Delta$$
$$\overline{G}\ \texttt{implements}\, I\langle\overline{W}\rangle \in \mathsf{sup}(R)$$
$$\Delta; \beta; I \vdash^?_{\mathrm{a}} \overline{T^?} \uparrow \overline{G} \twoheadrightarrow \overline{T}$$
$$(\forall i)\ W^?_i \sharp W_i$$

with $\mathcal{R} = \overline{T}\ \texttt{implements}\, I\langle\overline{W}\rangle$. We then have by (169) that $\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{G}$. The claim now follows with rule ENT-ALG-ENV.

- *Case* Rule ENT-NIL-ALG-IFACE$_1$: Then

$$\Delta;\beta;I \vdash_{\mathrm{a}} T \uparrow I\langle\overline{W}\rangle$$
$$1 \in \mathsf{pos}^+(I)$$
$$(\forall i)\ W_i^? \sharp W_i$$

with $\overline{T^?} = T$ and $\mathcal{R} = T\,\texttt{implements}\,I\langle\overline{W}\rangle$. The claim follows from rule ENT-ALG-IFACE$_1$.

- *Case* Rule ENT-NIL-ALG-IFACE$_2$: Then

$$1 \in \mathsf{pos}^+(J)$$
$$J\langle\overline{V}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$$
$$(\forall i)\ W_i^? \sharp W_i$$

with $\overline{T^?} = J\langle\overline{V}\rangle$ and $\mathcal{R} = J\langle\overline{V}\rangle\,\texttt{implements}\,I\langle\overline{W}\rangle$. The claim follows from rule ENT-ALG-IFACE$_2$.

- *Case* Rule ENT-NIL-ALG-IMPL: Then

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots$$
$$\Delta;\beta;I \vdash_{\mathrm{a}}^? \overline{T^?} \uparrow [\overline{U/X}]\overline{N} \twoheadrightarrow \overline{T}$$
$$(\forall i)\ W_i^? \sharp [\overline{U/X}]V_i$$
$$[\overline{U/X}]\overline{N}\,\texttt{implements}\,I\langle[\overline{U/X}]\overline{V}\rangle \notin \mathscr{G}$$
$$\Delta;\mathscr{G} \cup \{[\overline{U/X}]\overline{N}\,\texttt{implements}\,I\langle[\overline{U/X}]\overline{V}\rangle\};\texttt{false} \Vdash_{\mathrm{a}} [\overline{U/X}]\overline{P}$$

with $\mathcal{R} = \overline{T}\,\texttt{implements}\,I\langle[\overline{U/X}]\overline{V}\rangle$. From $\Delta;\beta;I \vdash_{\mathrm{a}}^? \overline{T^?} \uparrow [\overline{U/X}]\overline{N} \twoheadrightarrow \overline{T}$ we get with (169) that $\Delta;\beta;I \vdash_{\mathrm{a}} \overline{T} \uparrow [\overline{U/X}]\overline{N}$. The claim now follows with rule ENT-ALG-IMPL.

End case distinction on the last rule used in $\mathcal{D}$. □

Lemma 11.4. *If I is a single-headed interface, then $1 \in \mathsf{disp}(I)$.*

PROOF. The proof is by induction on the depth of I in the interface inheritance hierarchy. If the depth is 0 then I's definition has the form

$$\texttt{interface}\ I\langle\overline{X}\rangle\,[Y]\ \texttt{where}\ \overline{R}\,\{\ldots\ \mathit{rcsig}\,\}$$

It is now easy to verify that $1 \in \mathsf{disp}(I)$.

If the depth is $n > 0$ then I's definition has the form

$$\texttt{interface}\ I\langle\overline{X}\rangle\,[Y\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{S}\,\{\ldots\ \mathit{rcsig}\,\}$$

with $(\forall i)\ R_i = Y\,\texttt{implements}\,J_i\langle\overline{T_i}\rangle$ by criterion WF-IFACE-3 and the depth of each J_i is smaller then n. By the I.H., we have $(\forall i)\ 1 \in \mathsf{disp}(J_i)$. It is now easy to verify that $1 \in \mathsf{disp}(I)$. □

Lemma 11.5 (Completeness of entailment for nillable constraints)**.** *If $\Delta \Vdash_{\mathrm{a}} \overline{T}\,\texttt{implements}\,I\langle\overline{V}\rangle$ and $\overline{T^?}\,\overline{V^?} \sharp \overline{T}\,\overline{V}$ and $T_i^? \neq \mathsf{nil}$ for $i \in \mathsf{disp}(I)$, then $\Delta \Vdash_{\mathrm{a}}^? \overline{T^?}\,\texttt{implements}\,I\langle\overline{V^?}\rangle \twoheadrightarrow \overline{U}\,\texttt{implements}\,I\langle\overline{V}\rangle$ such that $\Delta \vdash_{\mathrm{a}}' T_i \leq U_i$ for all i and $U_i = T_i$ for those i with $T_i^? \neq \mathsf{nil}$ or $i \notin \mathsf{pos}^-(I)$.*

PROOF. We first show:

$$\begin{aligned}&\text{If } \overline{T^?}^n \sharp \overline{T}^n \text{ and } \Delta;\texttt{false};I \vdash_{\mathrm{a}} \overline{T}^n \uparrow \overline{U}^n \text{ then } \Delta;\texttt{false};I \vdash_{\mathrm{a}} \overline{T^?}^n \uparrow \overline{U}^n \twoheadrightarrow \overline{V}^n\\ &\quad\text{such that } \Delta \vdash_{\mathrm{a}}' T_i \leq V_i \text{ for all } i \text{ and}\\ &\quad V_i = T_i \text{ for those } i \text{ with } T_i^? \neq \mathsf{nil} \text{ or } i \notin \mathsf{pos}^-(I).\end{aligned} \tag{170}$$

{eq:lemma-aux::lemma:completeness-nil-entail}

Assume $\Delta;\mathscr{G};\texttt{false} \vdash_{\mathrm{a}} \overline{T}^n \uparrow \overline{U}^n$ and $\overline{T^?}^n \sharp \overline{T}^n$. By inverting rule ENT-ALG-LIFT, we get $\Delta \vdash_{\mathrm{a}}' T_i \leq U_i$ for all i and $T_i = U_i$ for $i \notin \mathsf{pos}^-(I)$. By rule ENT-NIL-ALG-LIFT, we have $\Delta;\mathscr{G};\texttt{false} \vdash_{\mathrm{a}} \overline{T^?}^n \uparrow \overline{U}^n \twoheadrightarrow \overline{V}^n$ for some $\overline{V}$. Now let $i \in [n]$.

- If $T_i^? = \mathsf{nil}$ then $V_i = U_i$. Hence, $\Delta \vdash_{\mathrm{a}} T_i \leq V_i$. If additionally $i \notin \mathsf{pos}^-(I)$, then $V_i = U_i = T_i$.
- If $T_i^? \neq \mathsf{nil}$ then $V_i = T_i^?$. With $T_i^? \sharp T_i$ then $V_i = T_i$.

This finishes the proof of (170).

We now show that $\mathcal{D} :: \Delta; \mathscr{G}; \mathtt{false} \Vdash_{\mathrm{a}} \overline{T}\, \mathtt{implements}\, I\langle\overline{V}\rangle$ and $\overline{T^?}\,\overline{V^?} \sharp \overline{T}\,\overline{V}$ and $T_i^? \neq \mathsf{nil}$ for $i \in \mathsf{disp}(i)$ imply $\Delta; \mathscr{G}; \mathtt{false} \Vdash_{\mathrm{a}}^? \overline{T^?}\, \mathtt{implements}\, I\langle\overline{V^?}\rangle \twoheadrightarrow \overline{U}\, \mathtt{implements}\, I\langle\overline{V}\rangle$ such that $\Delta \vdash_{\mathrm{a}}' T_i \leq U_i$ for all i and $U_i = T_i$ for those i with $T_i^? \neq \mathsf{nil}$ or $i \notin \mathsf{pos}^-(I)$. The claim then follows with rule ENT-NIL-ALG-MAIN.

Case distinction on the last rule used in $\mathcal{D}$.

- *Case* Rule ENT-ALG-ENV: Then

$$R \in \Delta$$
$$\overline{G}\, \mathtt{implements}\, I\langle\overline{V}\rangle \in \mathsf{sup}(R)$$
$$\Delta; \mathtt{false}; I \vdash_{\mathrm{a}} \overline{T} \uparrow \overline{G}$$

  By (170) we have $\Delta; \mathscr{G}; \beta \vdash_{\mathrm{a}}^? \overline{T^?} \uparrow \overline{G} \twoheadrightarrow \overline{U}$ such that $\overline{U}$ has the desired properties. The claim now follows by rule ENT-NIL-ALG-ENV.

- *Case* Rule ENT-ALG-IFACE$_1$: Then

$$\Delta; \mathtt{false}; I \vdash_{\mathrm{a}} T \uparrow I\langle\overline{V}\rangle$$
$$1 \in \mathsf{pos}^+(I)$$

  with $\overline{T} = T$. By Lemma 11.4, $1 \in \mathsf{disp}(I)$. Hence, $T_1^? = T_1 = T$. We get with (170) that $\Delta; \mathscr{G}; \beta \vdash_{\mathrm{a}}^? T^? \uparrow I\langle\overline{V}\rangle \twoheadrightarrow T$. The claim now follows by rule ENT-NIL-ALG-IFACE$_1$.

- *Case* Rule ENT-ALG-IFACE$_2$: Then

$$1 \in \mathsf{pos}^+(J)$$
$$J\langle\overline{W}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle$$

  with $\overline{T} = J\langle\overline{W}\rangle$. By Lemma 11.4, $1 \in \mathsf{disp}(I)$. Hence, $T_1^? = T_1 = J\langle\overline{W}\rangle$. The claim now follows by rule ENT-NIL-ALG-IFACE$_2$.

- *Case* Rule ENT-ALG-IMPL: Then

$$\mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\,\overline{N}\,]\ \mathtt{where}\ \overline{P}\ \ldots$$
$$\Delta; \beta; I \vdash_{\mathrm{a}} \overline{T} \uparrow [\overline{W/X}]\overline{N}$$
$$\overline{V} = [\overline{W/X}]\overline{V'}$$
$$[\overline{W/X}]\overline{N}\, \mathtt{implements}\, I\langle\overline{V}\rangle \notin \mathscr{G}$$
$$\Delta; \mathscr{G} \cup \{[\overline{W/X}]\overline{N}\, \mathtt{implements}\, I\langle\overline{V}\rangle\}; \mathtt{false} \Vdash_{\mathrm{a}} [\overline{W/X}]\overline{P}$$

  By (170), we have $\Delta; \beta; I \vdash_{\mathrm{a}}^? \overline{T^?} \uparrow [\overline{W/X}]\overline{N} \twoheadrightarrow \overline{U}$ such that $\overline{U}$ has the desired properties. The claim now follows by rule ENT-NIL-ALG-IMPL.

End case distinction on the last rule used in $\mathcal{D}$. □

Lemma 11.6. *If $\Delta \vdash T \leq G_1$ and $\Delta \vdash T \leq G_2$ then $\Delta \vdash G_1 \leq G_2$ or $\Delta \vdash G_2 \leq G_1$.*

PROOF. We first note that $\Delta \vdash T \leq G_i$ implies $\Delta \vdash_{\mathrm{q}}' T \leq G_i$ by Corollary 11.2. If $G_1 = \mathtt{Object}$ or $G_2 = \mathtt{Object}$, then the claim is obvious. Thus, assume $G_1 \neq \mathtt{Object}$ and $G_2 \neq \mathtt{Object}$.

Case distinction on the form of T.

- *Case* $T = X$ for some X: If $G_1 = X$ or $G_2 = X$ then the claim is obvious. Now assume $G_1 \neq X$ and $G_2 \neq X$. By Lemma 6.11 we have that

$$X\,\texttt{extends}\,G_i \in^+ \Delta \quad (i = 1,2) \tag{171}$$ {eq:inplus::lemma:multi-G-bounds-compat}

Define $\mathsf{level} : \mathit{TvarName} \to \mathbb{N}$ as follows. Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be a directed graph with

$$\mathscr{V} = \{X \in \mathit{TvarName} \mid X\,\texttt{extends}\,T \in \Delta \text{ or } Y\,\texttt{extends}\,X \in \Delta\}$$
$$\mathscr{E} = \{(X,Y) \mid Y\,\texttt{extends}\,X \in \Delta\}$$

Δ is contractive by criterion Wf-Tenv-2, so $\mathscr{G}$ is acyclic. Hence, there exists a topological ordering $X_0, X_1, \ldots, X_n$ on $\mathscr{V}$ such that $(X_i, X_j) \in \mathscr{E}$ implies $i < j$. Then

$$\mathsf{level}(X) = \begin{cases} i & \text{if } X \in \mathscr{V} \text{ and } X = X_i \\ 0 & \text{if } X \notin \mathscr{V} \end{cases}$$

We have that

$$X\,\texttt{extends}\,Y \in \Delta \text{ implies } \mathsf{level}(X) > \mathsf{level}(Y)$$

We now show that $X\,\texttt{extends}\,G_i \in^+ \Delta$ for $i = 1,2$ implies $\Delta \vdash_q{}' G_1 \leq G_2$ or $\Delta \vdash_q{}' G_2 \leq G_1$ by induction on $\mathsf{level}(X)$. Together with (171), this finishes the case "$T = X$".

  - $\mathsf{level}(X) = 0$. Assume $X\,\texttt{extends}\,Y \in \Delta$. Then $0 = \mathsf{level}(X) > \mathsf{level}(Y)$ which is impossible because $\mathsf{level}(Y) \in \mathbb{N}$.
    Hence, $G_i = N_i$ for some N_i and $X\,\texttt{extends}\,G_i \in \Delta$ (for $i = 1,2$). The claim now follows with criterion Wf-Tenv-4.
  - $\mathsf{level}(X) = n > 0$ and the claim holds for $n' < n$.
    Case distinction on the last rules in the derivations of $X\,\texttt{extends}\,G_i \in^+ \Delta$.
    * *Case* $\in^+$-direct / $\in^+$-direct: The claim follows with criterion Wf-Tenv-4.
    * *Case* $\in^+$-step / $\in^+$-direct: Then

$$\begin{gathered} X\,\texttt{extends}\,Y \in \Delta \\ Y\,\texttt{extends}\,G_1 \in^+ \Delta \\ X\,\texttt{extends}\,G_2 \in \Delta \end{gathered} \tag{172}$$ {eq:1::lemma:multi-G-bounds-compat}

      By criterion Wf-Tenv-4 either $\Delta \vdash Y \leq G_2$ or $\Delta \vdash G_2 \leq Y$. By Corollary 11.2 either $\Delta \vdash_q{}' Y \leq G_2$ or $\Delta \vdash_q{}' G_2 \leq Y$.
      · Suppose $\Delta \vdash_q{}' Y \leq G_2$. If $Y = G_2$ then $\Delta \vdash G_2 \leq G_1$ by (172) and Lemma 6.9. If $Y \neq G_2$ then $Y\,\texttt{extends}\,G_2 \in^+ \Delta$ by Lemma 6.11. Because $\mathsf{level}(Y) < \mathsf{level}(X)$ we can use the I.H. on (172) and get the desired result.
      · Suppose $\Delta \vdash_q{}' G_2 \leq Y$. By Lemma 6.11, $G_2 = Z$ for some Z with either $Y = Z$ or $Z\,\texttt{extends}\,Y \in^+ \Delta$. If $Y = Z = G_2$ then $\Delta \vdash G_2 \leq G_1$ by (172) and Lemma 6.9. Otherwise, $Z\,\texttt{extends}\,G_1 \in^+ \Delta$ by (172) and Lemma 6.8, so $\Delta \vdash G_2 \leq G_1$ by Lemma 6.9.
    * *Case* $\in^+$-direct / $\in^+$-step: Analogously to the preceding case.
    * *Case* $\in^+$-step / $\in^+$-step: Then

$$\begin{gathered} X\,\texttt{extends}\,Y_1 \in \Delta \\ Y_1\,\texttt{extends}\,G_1 \in^+ \Delta \end{gathered} \tag{173}$$ {eq:2::lemma:multi-G-bounds-compat}

$$\begin{gathered} X\,\texttt{extends}\,Y_2 \in \Delta \\ Y_2\,\texttt{extends}\,G_2 \in^+ \Delta \end{gathered} \tag{174}$$ {eq:3::lemma:multi-G-bounds-compat}

By criterion WF-TENV-4 either $\Delta \vdash Y_1 \leq Y_2$ or $\Delta \vdash Y_2 \leq Y_1$. We now consider the case $\Delta \vdash Y_1 \leq Y_2$, the proof for the other case is very similar. From $\Delta \vdash Y_1 \leq Y_2$ we get $\Delta \vdash_q' Y_1 \leq Y_2$ by Corollary 11.2. With Lemma 6.11 either $Y_1 = Y_2$ or $Y_1\ \texttt{extends}\ Y_2 \in^+ \Delta$. In the following, note that $\mathsf{level}(Y_i) < \mathsf{level}(X)$ for $i = 1, 2$.

- If $Y_1 = Y_2$ then the claim follows by applying the I.H. to (173) and (174).
- If $Y_1\ \texttt{extends}\ Y_2 \in^+ \Delta$, then we get by the (173) and the I.H. that either $\Delta \vdash Y_2 \leq G_1$ or $\Delta \vdash G_1 \leq Y_2$. In the latter case, we have with $Y_2\ \texttt{extends}\ G_2 \in^+ \Delta$, Lemma 6.9, and transitivity that $\Delta \vdash G_1 \leq G_2$. If $\Delta \vdash Y_2 \leq G_1$ then $\Delta \vdash_q' Y_2 \leq G_1$ by Corollary 11.2. With Lemma 6.11 either $Y_2 = G_1$ or $Y_2\ \texttt{extends}\ G_1 \in^+ \Delta$. In the former case, we get with (174) and Lemma 6.9 that $\Delta \vdash G_1 \leq G_2$. In the latter case, the claim follows by applying the I.H. to $Y_2\ \texttt{extends}\ G_1 \in^+ \Delta$ and (174).

End case distinction on the last rules in the derivations of $X\ \texttt{extends}\ G_i \in^+ \Delta$.

- *Case* $T = N$ for some N or $T = K$ for some K: Because $\Delta \vdash_q' T \leq G_i$ and $G_i \neq \texttt{Object}$ we have with Lemma 6.11 that $T = N$ and $G_i = N_i$ ($i = 1, 2$). Hence, $N \trianglelefteq_c N_1$ and $N \trianglelefteq_c N_2$. The claim now follows by Lemma 7.26.

End case distinction on the form of T. □

Lemma 11.7 (Antisymmetry of kernel subtyping)**.** *If $\Delta \vdash_q' T \leq U$ and $\Delta \vdash_q' U \leq T$ then $T = U$.*

PROOF. We proceed by case distinction on the last rules of the two derivations. The only combinations possible are:

SUB-Q-ALG-OBJ / SUB-Q-ALG-OBJ: Then $T = \texttt{Object} = U$.

SUB-Q-ALG-OBJ / SUB-Q-ALG-CLASS or SUB-Q-ALG-CLASS / SUB-Q-ALG-OBJ: Impossible because programs cannot define Object.

SUB-Q-ALG-VAR-REFL / SUB-Q-ALG-VAR-REFL: Then $T = X = U$ for some X.

SUB-Q-ALG-VAR / SUB-Q-ALG-VAR: Then $T = X$, $X\ \texttt{extends}\ T' \in \Delta$, and $U = Y$, $Y\ \texttt{extends}\ U' \in \Delta$, and $\Delta \vdash_q' T' \leq Y$, $\Delta \vdash_q' U' \leq X$. By Lemma 6.11 then $T' = Y'$, $Y'\ \texttt{extends}\ Y \in^* \Delta$, and $U' = X'$, $X'\ \texttt{extends}\ X \in^* \Delta$. Hence, we have $X\ \texttt{extends}\ Y' \in \Delta$, $Y'\ \texttt{extends}\ Y \in^* \Delta$, $Y\ \texttt{extends}\ X' \in \Delta$, and $X'\ \texttt{extends}\ X \in^* \Delta$. This is a contradiction because Δ is contractive by criterion WF-TENV-2.

SUB-Q-ALG-CLASS / SUB-Q-ALG-CLASS: Then $T = N_1$, $U = N_2$ with $N_1 \trianglelefteq_c N_2$ and $N_2 \trianglelefteq_c N_1$. Because the class graph is acyclic by criterion WF-PROG-5 we have $N_1 = N_2$.

SUB-Q-ALG-IFACE / SUB-Q-ALG-IFACE: Then $T = K_1$, $U = K_2$ with $K_1 \trianglelefteq_i K_2$ and $K_2 \trianglelefteq_i K_1$. Because the interface graph is acyclic by criterion WF-PROG-5 we have $K_1 = K_2$. □

Lemma 11.8. *If $\Delta \vdash_q' \texttt{Object} \leq T$ then $T = \texttt{Object}$.*

PROOF. With rule SUB-Q-ALG-OBJ, we have $\Delta \vdash_q' T \leq \texttt{Object}$. The claim now follows with Lemma 11.7. □

Lemma 11.9 (Existence of $\sqcap$)**.** *If $\Delta \vdash T \leq G_i$ for $i = 1, 2$ then there exists H with $\Delta \vdash G_1 \sqcap G_2 = H$.*

PROOF. With Lemma 11.6 we have either $\Delta \vdash G_1 \leq G_2$ or $\Delta \vdash G_2 \leq G_1$. With rule GLB-LEFT or GLB-RIGHT, respectively, we then have $\Delta \vdash G_1 \sqcap G_2 = G_1$ or $\Delta \vdash G_1 \sqcap G_2 = G_2$. □

Lemma 11.10. *If $\Delta \vdash N_1 \sqcap N_2 = H$ then $\Delta' \vdash N_1 \sqcap N_2 = H$ for any Δ'.*

PROOF. From $\Delta \vdash N_1 \sqcap N_2 = H$ we have w.l.o.g. $N_1 \trianglelefteq_c N_2$. Hence, $\Delta' \vdash N_1 \leq N_2$, so the claim holds. □

Lemma 11.11. *If $\Delta \Vdash \overline{T}\,\texttt{implements}\,I\langle\overline{U}\rangle$ and $\Delta \Vdash \overline{V}\,\texttt{implements}\,I\langle\overline{W}\rangle$ such that for all $i \in \mathsf{disp}(I)$ there exists T_i' with $\Delta \vdash_{\mathrm{a}}{}' T_i' \leq T_i$ and $\Delta \vdash_{\mathrm{a}}{}' T_i' \leq V_i$, then $\overline{U} = \overline{W}$ and $T_j = V_j$ for all $j \notin \mathsf{disp}(I) \cup \mathsf{pos}^{-}(I)$.*

Proof. Define $\mathcal{P} = \Delta \Vdash \overline{T}\,\texttt{implements}\,I\langle\overline{U}\rangle$ and $\mathcal{Q} = \Delta \Vdash \overline{V}\,\texttt{implements}\,I\langle\overline{W}\rangle$. We first prove the following auxiliary lemma:

$$\begin{gathered}\textit{If } \Delta \Vdash_{\mathrm{q}}{}' \mathcal{P} \textit{ and } \Delta \Vdash_{\mathrm{q}}{}' \mathcal{Q} \textit{ and for all } i \in \mathsf{disp}(I) \textit{ there exists } T_i' \textit{ with}\\ \Delta \vdash_{\mathrm{q}}{}' T_i' \leq T_i \textit{ and } \Delta \vdash_{\mathrm{q}}{}' T_i' \leq V_i, \textit{ then } \overline{U} = \overline{W} \textit{ and } T_j = V_j\\ \textit{for all } j \notin \mathsf{disp}(I) \cup \mathsf{pos}^{-}(I).\end{gathered} \tag{175}$$ {eq:sublemma::lemma:entails-fixed}

The proof is by induction in the combined height of the derivations of $\Delta \Vdash_{\mathrm{q}}{}' \mathcal{P}$ and $\Delta \Vdash_{\mathrm{q}}{}' \mathcal{Q}$. We proceed by case analysis on the last rules of these derivations. The following table lists all possible cases; cases marked with $\lightning$ can never occur because they put conflicting constraints on the form of $\mathcal{P}$ and $\mathcal{Q}$. The remaining cases are dealt with shortly.

$\Delta \Vdash_{\mathrm{q}}{}' \mathcal{P}$ \ $\Delta \Vdash_{\mathrm{q}}{}' \mathcal{Q}$	ENT-Q-ALG-ENV	ENT-Q-ALG-IMPL	ENT-Q-ALG-IFACE
ENT-Q-ALG-ENV	(1)	(2)	$\lightning$
ENT-Q-ALG-IMPL	(2)	(3)	$\lightning$
ENT-Q-ALG-IFACE	$\lightning$	$\lightning$	(4)

For (1), (2), and (3) we have $\overline{T} = \overline{G}$ and $\overline{V} = \overline{G'}$ for some $\overline{G}$ and $\overline{G'}$. Hence, by Lemma 11.9:

$$\textit{for all } i \in \mathsf{disp}(I) \textit{ exists } H_i \textit{ with } \Delta \vdash G_i \sqcap G_i' = H_i \tag{176}$$ {eq:glb-ex::lemma:entails-fixed}

1. Then $\mathcal{P} \in \mathsf{sup}(\Delta)$ and $\mathcal{Q} \in \mathsf{sup}(\Delta)$. The claim now follows with criterion Wf-Tenv-7.
2. Then, w.l.o.g., $\mathcal{P} \in \mathsf{sup}(\Delta)$ and $\mathcal{Q} = [\overline{U'/X}](\overline{N}\,\texttt{implements}\,I\langle\overline{W'}\rangle)$ for some

   $$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{W'}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots$$

   As in the preceding case, the claim follows with criterion Wf-Tenv-7.
3. Then

   $$\begin{gathered}\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{U'}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots\\ \texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{W'}\rangle\ [\,\overline{M}\,]\ \texttt{where}\ \overline{Q}\ \ldots\end{gathered}$$

   such that $\mathcal{P} = \sigma(\overline{N}\,\texttt{implements}\,I\langle\overline{U'}\rangle)$ with $\mathsf{dom}(\sigma) = \overline{X}$ and $\mathcal{Q} = \tau(\overline{M}\,\texttt{implements}\,I\langle\overline{W'}\rangle)$ with $\mathsf{dom}(\tau) = \overline{Y}$. We have by (176) and Lemma 11.10 that

   $$\text{for all } i \in \mathsf{disp}(I) \text{ exists } H_i \text{ with } \emptyset \vdash \sigma N_i \sqcap \tau M_i = H_i$$

   The claim now follows with criterion Wf-Prog-2.
4. Then $\overline{T} = J\langle\overline{U'}\rangle$, $1 \in \mathsf{pos}^{+}(J)$, $J\langle\overline{U'}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{U}\rangle$, and $\overline{V} = J'\langle\overline{W'}\rangle$, $1 \in \mathsf{pos}^{+}(J')$, $J'\langle\overline{W'}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$. Because I is a single-headed interface, $1 \in \mathsf{disp}(I)$ by Lemma 11.4. Hence,

   $$\begin{gathered}\Delta \vdash_{\mathrm{a}}{}' T_1' \leq J\langle\overline{U'}\rangle\\ \Delta \vdash_{\mathrm{a}}{}' T_1' \leq J'\langle\overline{W'}\rangle\end{gathered}$$

   By Lemma 6.11 one of the following holds:

   - $T_1' = X$ and $X\,\texttt{extends}\,K \in^{+} \Delta$ with $K \trianglelefteq_{\mathrm{i}} J\langle\overline{U'}\rangle$ and $X\,\texttt{extends}\,K' \in^{+} \Delta$ with $K' \trianglelefteq_{\mathrm{i}} J'\langle\overline{W'}\rangle$. With Lemma 6.9 and Lemma 6.5 then $\Delta \vdash_{\mathrm{a}}{}' X \leq I\langle\overline{U}\rangle$ and $\Delta \vdash_{\mathrm{a}}{}' X \leq I\langle\overline{W}\rangle$. Criterion Wf-Tenv-5 now yields $\overline{U} = \overline{W}$. as required.

- $T_1' = L$ with $L \trianglelefteq_i J\langle\overline{U'}\rangle$ and $L \trianglelefteq_i J'\langle\overline{W'}\rangle$. With Lemma 6.2 then $L \trianglelefteq_i I\langle\overline{U}\rangle$ and $L \trianglelefteq_i I\langle\overline{W}\rangle$. Hence, $\overline{U} = \overline{W}$ by criterion WF-PROG-7.

This finishes the proof of (175).

From $\Delta \Vdash \mathcal{P}$ and $\Delta \Vdash \mathcal{Q}$ we have $\Delta \Vdash_q \mathcal{P}$ and $\Delta \Vdash_q \mathcal{Q}$. By Lemma 6.27 there exists $\overline{T''}$ and $\overline{V'}$ such that for all i

$$\Delta \vdash_q{}' T_i \leq T_i''$$
$$T_i = T_i'' \text{ if } i \notin \mathsf{pos}^-(I)$$
$$\Delta \Vdash_q{}' \overline{T''}\,\texttt{implements}\,I\langle\overline{U}\rangle$$
$$\Delta \vdash_q{}' V_i \leq V_i'$$
$$V_i = V_i' \text{ if } i \notin \mathsf{pos}^-(I)$$
$$\Delta \Vdash_q{}' \overline{V'}\,\texttt{implements}\,I\langle\overline{W}\rangle$$

With Lemma 6.5 then $\Delta \vdash_a{}' T_i' \leq T_i''$ and $\Delta \vdash_a{}' T_i' \leq V_i'$ for all $i \in \mathsf{disp}(I)$. With (175) now $\overline{U} = \overline{W}$ and $T_i'' = V_i'$ if $i \notin \mathsf{disp}(I) \cup \mathsf{pos}^-(I)$. Assume $i \notin \mathsf{disp}(I) \cup \mathsf{pos}^-(I)$. Then $i \notin \mathsf{pos}^-(I)$, so $T_i = T_i''$ and $V_i = V_i'$. Hence, $T_i = V_i$ for $i \notin \mathsf{disp}(I) \cup \mathsf{pos}^-(I)$. □

Lemma 11.12 (Finiteness of set of kernel supertypes)**.** *The set* $\{U \mid \Delta \vdash_q{}' T \leq U\}$ *is finite for any* T *and* Δ*.*

PROOF. We prove that there exists a bound on the size of all types $U \in \{U \mid \Delta \vdash_q{}' T \leq U\}$. Then, because the set of types of a certain size is finite, $\{U \mid \Delta \vdash_q{}' T \leq U\}$ must be finite.

Let $\delta \in \mathbb{N}$ be a bound on the size of Δ and the program's superclasses and superinterfaces. That is,

- if $P \in \Delta$ then $\mathsf{size}(P) \leq \delta$,
- if $\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \texttt{where}\ \overline{P}\ \ldots$ then $\mathsf{size}(N) \leq \delta$,
- if $\texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\,\overline{R}]\ \ldots$ then $\mathsf{size}(\overline{R}) \leq \delta$.

Define the *weight of a type* as follows:

$$\begin{aligned}
&\mathsf{weight}(X) = \max\{\mathsf{weight}(T) \mid X\ \texttt{extends}\,T \in \Delta\}\\
&\mathsf{weight}(N) = \mathsf{size}(N)\\
&\mathsf{weight}(K) = \mathsf{size}(K)
\end{aligned}$$

(We use the convention that $\max\emptyset = 1$.) The definition of weight is well-formed (*i.e.* terminating) because Δ is contractive by criterion WF-TENV-2. Moreover, $\mathsf{weight}(T) \in \mathbb{N}^+$ and $\mathsf{weight}(T) \geq \mathsf{size}(T)$ for all types T.

Define the *level* of a type as follows:

$$\begin{aligned}
\mathsf{level}(\texttt{Object}) &= 1\\
\mathsf{level}(C\langle\overline{T}\rangle) &= n+1 && \text{if } \texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \ldots \text{ and } \mathsf{level}([\overline{T/X}]N) = n\\
\mathsf{level}(I\langle\overline{T}\rangle) &= 1 && \text{if } \texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}]\ \ldots\\
\mathsf{level}(I\langle\overline{T}\rangle) &= n+1 && \text{if } \texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\,\overline{R}]\ \ldots,\\
& && \quad R_i = \overline{V_i}\,\texttt{implements}\,K_i, \text{ and}\\
& && \quad n = \max_i(\mathsf{level}([\overline{T/X}]K_i))\\
\mathsf{level}(X) &= \max\{\mathsf{level}(T) \mid X\ \texttt{extends}\,T \in \Delta\}
\end{aligned}$$

The definition of level is well-formed (*i.e.*, terminating) because the class and interface graph is acyclic by criterion WF-PROG-5. Moreover, $\mathsf{level}(T) \in \mathbb{N}^+$ for all types T. We now show that

$$\Delta \vdash_q{}' T \leq U \textit{ implies } \mathsf{weight}(U) \leq \delta^{\mathsf{level}(T)} \cdot \mathsf{weight}(T) \tag{177}$$ {eq:weight-dec::lemma:finite-supertypes}

The proof of (177) is by induction on the derivation of $\Delta \vdash_q{}' T \leq U$.

Case distinction on the last rule used in the derivation of $\Delta \vdash_q{}' T \leq U$.

- *Case* SUB-Q-ALG-OBJ: Obvious.
- *Case* SUB-Q-ALG-VAR-REFL: Obvious.
- *Case* SUB-Q-ALG-VAR: Then $T = X$ and

$$\frac{X\ \texttt{extends}\ T' \in \Delta \qquad \Delta \vdash_{\mathrm{q}}{}' T' \leq U}{\Delta \vdash_{\mathrm{q}}{}' X \leq U}$$

  By the I.H. $\mathsf{weight}(U) \leq \delta^{\mathsf{level}(T')} \cdot \mathsf{weight}(T') \leq \delta^{\mathsf{level}(X)} \cdot \mathsf{weight}(X)$.

- *Case* SUB-Q-ALG-CLASS: Then $T = N$, $U = N'$, and $N \trianglelefteq_{\mathrm{c}} N'$. We now show that

$$N \trianglelefteq_{\mathrm{c}} N' \textit{ implies } \mathsf{size}(N') \leq \delta^{\mathsf{level}(N)} \cdot \mathsf{size}(N) \tag{178}$$ {eq:size-dec-N::lemma:finite-supertypes}

  We then have $\mathsf{weight}(N') = \mathsf{size}(N') \leq \delta^{\mathsf{level}(N)} \cdot \mathsf{size}(N) = \delta^{\mathsf{level}(N)} \cdot \mathsf{weight}(N)$ as required. The proof of (178) is by induction on the derivation of $N \trianglelefteq_{\mathrm{c}} N'$.

  Case distinction on the last rule used in the derivation of $N \trianglelefteq_{\mathrm{c}} N'$.

  - *Case* EXT-C-REFL: Obvious.
  - *Case* EXT-C-SUPER: Then $N = C\langle\overline{T}\rangle$ and

$$\frac{\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ M \ldots \qquad [\overline{T/X}]M \trianglelefteq_{\mathrm{c}} N'}{C\langle\overline{T}\rangle \trianglelefteq_{\mathrm{c}} N'}$$

    We have

$$\begin{aligned}
\mathsf{size}([\overline{T/X}]M) &\leq \mathsf{size}(M) + \mathsf{max}_i(\mathsf{size}(T_i)) \cdot (\mathsf{size}(M) - 1) \\
&\leq \mathsf{size}(M) + (\mathsf{size}(N) - 1) \cdot (\mathsf{size}(M) - 1) \\
&= \mathsf{size}(N) \cdot \mathsf{size}(M) - \mathsf{size}(N) + 1 \\
&\leq \delta \cdot \mathsf{size}(N) \\
\mathsf{level}(N) &= \mathsf{level}([\overline{T/X}]M) + 1
\end{aligned}$$

    Hence,

$$\begin{aligned}
\mathsf{size}(N') &\overset{I.H.}{\leq} \delta^{\mathsf{level}([\overline{T/X}]M)} \cdot \mathsf{size}([\overline{T/X}]M) \\
&\leq \delta^{\mathsf{level}([\overline{T/X}]M)} \cdot \delta \cdot \mathsf{size}(N) = \delta^{\mathsf{level}(N)} \cdot \mathsf{size}(N)
\end{aligned}$$

  End case distinction on the last rule used in the derivation of $N \trianglelefteq_{\mathrm{c}} N'$.

- *Case* SUB-Q-ALG-IFACE: Hence, $T = K$, $U = K'$, and $K \trianglelefteq_{\mathrm{i}} K'$. Similar to the preceding case, we show that $K \trianglelefteq_{\mathrm{i}} K'$ implies $\mathsf{size}(K') \leq \delta^{\mathsf{level}(K)} \cdot \mathsf{size}(K)$ by induction on the derivation of $K \trianglelefteq_{\mathrm{i}} K'$. The claim also follows analogously to the preceding case.

End case distinction on the last rule used in the derivation of $\Delta \vdash_{\mathrm{q}}{}' T \leq U$. □

Lemma 11.13. *If $T \in \mathsf{MUB}_\Delta(\mathscr{U})$ then $\Delta \vdash_{\mathrm{a}}{}' U \leq T$ for all $U \in \mathscr{U}$.*

PROOF. Obvious. □

Lemma 11.14. *Let $\mathscr{T}$ be a non-empty set of types. Suppose $\Delta \vdash_{\mathrm{a}}{}' T \leq V$ for all $T \in \mathscr{T}$. Then there exists a $V' \in \mathsf{MUB}_\Delta(\mathscr{T})$ such that $\Delta \vdash_{\mathrm{a}}{}' V' \leq V$.*

Proof. We argue by contradiction. To do so, we construct an infinite chain $U_0, U_1, U_2, \dots$ such that $U_i \neq U_j$ for all $i \neq j$ and $\Delta \vdash_a{}' T \leq U_i$ for all $T \in \mathscr{T}$ and all i. Hence, because $\mathscr{T} \neq \emptyset$, there exists some $T \in \mathscr{T}$ such that the set $\{U \mid \Delta \vdash_a{}' T \leq U\}$ is infinite. This is then a contradiction to Lemma 11.12.

Here is how we construct the infinite chain $U_0, U_1, U_2, \dots$:

- Assume $V = U_0 \notin \mathsf{MUB}_\Delta(\mathscr{T})$. (Otherwise, choose $V' = U_0$ and we are done.) Hence, there exists $U_1 \neq U_0$ with $\Delta \vdash_a{}' T \leq U_1$ for all $T \in \mathscr{T}$ and $\Delta \vdash_a{}' U_1 \leq U_0$.
- Assume $U_1 \notin \mathsf{MUB}_\Delta(\mathscr{T})$. (Otherwise, choose $V' = U_1$ and we are done.) Hence, there exists $U_2 \neq U_1$ with $\Delta \vdash_a{}' T \leq U_2$ for all $T \in \mathscr{T}$ and $\Delta \vdash_a{}' U_2 \leq U_1$.
- ...
- Assume $U_i \notin \mathsf{MUB}_\Delta(\mathscr{T})$. (Otherwise, choose $V' = U_i$ and we are done.) Hence, there exists $U_{i+1} \neq U_i$ with $\Delta \vdash_a{}' T \leq U_{i+1}$ for all $T \in \mathscr{T}$ and $\Delta \vdash_a{}' U_{i+1} \leq U_i$.
- ...

From this construction we have:

$$\begin{gathered} \Delta \vdash_a{}' T \leq U_i \quad \text{for all } i \in \mathbb{N}, T \in \mathscr{T} \\ U_i \neq U_{i+1} \quad \text{for all } i \in \mathbb{N} \\ \Delta \vdash_a{}' U_{i+1} \leq U_i \quad \text{for all } i \in \mathbb{N} \end{gathered}$$

We still have to verify that $U_i \neq U_j$ if $i \neq j$. Suppose $i < j$ with $U_i = U_j$. Because subtyping is transitive we have $\Delta \vdash_a{}' U_j \leq U_{i+1}$. Hence, $\Delta \vdash_a{}' U_i \leq U_{i+1}$. But we also have $\Delta \vdash_a{}' U_{i+1} \leq U_i$. With Lemma 11.7 now $U_i = U_{i+1}$ which is a contradiction. □

If we choose $V = \texttt{Object}$ in Lemma 11.14, then we get the following corollary:

Corollary 11.15. *For any set of types $\mathscr{T} \neq \emptyset$, $\mathsf{MUB}_\Delta(\mathscr{T}) \neq \emptyset$.*

Lemma 11.16. *Let $\mathscr{T}$ be a non-empty set of types. If $G_1 \in \mathsf{MUB}_\Delta(\mathscr{T})$ and $G_2 \in \mathsf{MUB}_\Delta(\mathscr{T})$ then $G_1 = G_2$.*

Proof. Because $\mathscr{T} \neq \emptyset$, there exists $T \in \mathscr{T}$ such that $\Delta \vdash_a{}' T \leq G_i$ for $i = 1, 2$. By Lemma 11.6 either $\Delta \vdash_a{}' G_1 \leq G_2$ or $\Delta \vdash_a{}' G_2 \leq G_1$. W.l.o.g. assume $\Delta \vdash_a{}' G_1 \leq G_2$. But because $G_2 \in \mathsf{MUB}_\Delta(\mathscr{T})$ we must have that $G_1 = G_2$. □

Lemma 11.17. *If $\mathsf{bound}_\Delta(T) = N$ then $\Delta \vdash T \leq N$.*

Proof. Obvious. □

Lemma 11.18. *If $\Delta \vdash_a{}' T \leq N$ then $\mathsf{bound}_\Delta(T) = M$ with $M \trianglelefteq_c N$.*

Proof. Follows with a straightforward case distinction on the form of T. □

Lemma 11.19.

(i) If $N \trianglelefteq_c N'$ then $\mathsf{ftv}(N') \subseteq \mathsf{ftv}(N)$.

(ii) If $K \trianglelefteq_i K'$ then $\mathsf{ftv}(K') \subseteq \mathsf{ftv}(K)$.

(iii) If $\Delta \vdash_q{}' T \leq U$ then $\mathsf{ftv}(U) \subseteq \mathsf{ftv}(\Delta, T)$.

Proof. We prove all three parts by straightforward inductions on the given derivations. □

Lemma 11.20 (Strengthening)**.** *Let $\Delta' = \Delta, X$ $\texttt{implements}$ K and $\Delta'' = \Delta, X$.*

(*i*) *If* $\Delta' \vdash T$ ok *and* $X \notin \mathsf{ftv}(\Delta, K, T)$ *then* $\Delta \vdash T$ ok.

(*ii*) *If* $\Delta' \vdash \mathcal{P}$ ok *and* $X \notin \mathsf{ftv}(\Delta, K, \mathcal{P})$ *then* $\Delta \vdash \mathcal{P}$ ok.

(*iii*) *If* $\Delta'' \vdash T$ ok *and* $X \notin \mathsf{ftv}(\Delta, T)$ *then* $\Delta \vdash T$ ok.

(*iv*) *If* $\Delta'' \vdash \mathcal{P}$ ok *and* $X \notin \mathsf{ftv}(\Delta, \mathcal{P})$ *then* $\Delta \vdash \mathcal{P}$ ok.

PROOF. We first prove:

(*a*) *If* $\mathcal{D}_1 :: \Delta' \vdash_{\mathrm{q}}' V \leq U$ *then* $\Delta \vdash_{\mathrm{q}}' V \leq U$.

(*b*) *If* $\mathcal{D}_2 :: \Delta' \Vdash_{\mathrm{q}}' \mathcal{P}$ *and* $X \notin \mathsf{ftv}(\Delta, K, \mathcal{P})$ *then* $\Delta \Vdash_{\mathrm{q}}' \mathcal{P}$.

(*c*) *If* $\mathcal{D}_3 :: \Delta' \vdash_{\mathrm{q}} V \leq U$ *and* $X \notin \mathsf{ftv}(\Delta, V, U)$ *then* $\Delta \vdash_{\mathrm{q}} V \leq U$.

(*d*) *If* $\mathcal{D}_4 :: \Delta' \Vdash_{\mathrm{q}} \mathcal{P}$ *and* $X \notin \mathsf{ftv}(\Delta, K, \mathcal{P})$ *then* $\Delta \Vdash_{\mathrm{q}} \mathcal{P}$.

The proof of (a) is straightforward because kernel subtyping does not use `implements` constraints.

The proof of (b), (c), and (d) is by induction on the combined height of $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$.

(b) *Case distinction* on the last rule of the derivation of $\Delta' \Vdash_{\mathrm{q}}' \mathcal{P}$.

- *Case* rule ENT-Q-ALG-ENV: Then $R \in \Delta'$ and $\mathcal{P} \in \mathsf{sup}(R)$. Assume $R = X\ \texttt{implements}\ K$. By Lemma 6.23 we have $\mathcal{P} = X\ \texttt{implements}\ K'$. But this is a contradiction to the assumption $X \notin \mathsf{ftv}(\mathcal{P})$. Hence, $R \neq X\ \texttt{implements}\ K$, so $R \in \Delta$ and the claim follows with ENT-Q-ALG-ENV.
- *Case* rule ENT-Q-ALG-IMPL: Then

$$\frac{\texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{T}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots \qquad \Delta' \Vdash_{\mathrm{q}} [\overline{U/Y}]\overline{P}}{\Delta' \Vdash_{\mathrm{q}}' \underbrace{[\overline{U/Y}](\overline{N}\ \texttt{implements}\ I\langle\overline{T}\rangle)}_{=\mathcal{P}}}$$

  With criterion WF-IMPL-1 we have $\overline{X} \subseteq \mathsf{ftv}(\overline{N})$. With $X \notin \mathsf{ftv}(\mathcal{P})$ we then have $X \notin \mathsf{ftv}(\overline{U})$. Hence, $X \notin \mathsf{ftv}([\overline{U/X}]\overline{P})$. Applying part (d) of the I.H. yields $\Delta \Vdash_{\mathrm{q}} [\overline{U/X}]\overline{P}$, so the claim follows with ENT-Q-ALG-IMPL.
- *Case* rule ENT-Q-ALG-IFACE: Obvious.

End case distinction on the last rule of the derivation of $\Delta' \Vdash_{\mathrm{q}}' \mathcal{P}$.

(c) If the last rule of $\mathcal{D}_3$ is SUB-Q-ALG-KERNEL, then the claim follows by (a). Otherwise, we have

$$\Delta' \vdash_{\mathrm{q}}' V \leq W$$
$$\Delta' \Vdash_{\mathrm{q}}' W\ \texttt{implements}\ L$$

with $U = L$. By (a) then $\Delta \vdash_{\mathrm{q}}' V \leq W$. With Lemma 11.19 we have $\mathsf{ftv}(W) \subseteq \mathsf{ftv}(V, \Delta)$. Hence, $X \notin \mathsf{ftv}(W)$. With part (b) of the I.H. we then have $\Delta \Vdash_{\mathrm{q}}' W\ \texttt{implements}\ L$. The claim now follows with rule SUB-Q-ALG-IMPL.

(d) Follows trivially from (a) and parts (b), (c) of the I.H.

Next, we prove:

$$\textit{If } \Delta'' \Vdash_{\mathrm{q}} \mathcal{P} \textit{ then } \Delta \Vdash_{\mathrm{q}} \mathcal{P} \tag{179}$$ {eq:claim5::lemma:strengthening}

This claim holds trivially because constraint entailment does not use the type variable component of Δ'' at all.

Using (d) and (179), we easily show the original claim by an induction on the given derivations.
□

Lemma 11.21 (Interface inheritance propagates well-formedness). *If* $K \trianglelefteq_{\mathrm{i}} L$ *and* $\Delta \vdash K$ ok *then* $\Delta \vdash L$ ok

Proof. We proceed by induction on the derivation of $K \trianglelefteq_{\mathrm{i}} L$
Case distinction on the last rule of the derivation of $K \trianglelefteq_{\mathrm{i}} L$.

- *Case* rule EXT-I-REFL: Obvious.
- *Case* rule EXT-I-SUPER: Then

$$\frac{\texttt{interface}\, I\langle\overline{X}\rangle\,[Y\,\texttt{where}\,\overline{R}]\ \texttt{where}\ \overline{P}\ \ldots \qquad R_i = Y\,\texttt{implements}\,K' \qquad [\overline{V/X}]K' \trianglelefteq_{\mathrm{i}} L}{\Delta \vdash I\langle\overline{V}\rangle \leq L}$$

with $K = I\langle\overline{V}\rangle$. We now prove that $\Delta \vdash [\overline{V/X}]K'$ ok. The original claim then follows by the I.H.

Because $\Delta \vdash K$ ok, we have

$$\Delta, Y\,\texttt{implements}\,I\langle\overline{V}\rangle, Y \Vdash [\overline{V/X}]\overline{R}, \overline{P}$$
$$\Delta \vdash \overline{V}\ \mathsf{ok}$$

with

$$Y \notin \mathsf{ftv}(\overline{V}, \Delta) \tag{180}$$ {eq:y-fresh::lemma:cache-subset-goalset+cand}

Lemma 7.2 gives us $\Delta, Y\,\texttt{implements}\,I\langle\overline{V}\rangle, Y \vdash \overline{V}$ ok. The underlying program is well-typed, so $\overline{R}, \overline{P}, \overline{X}, Y \vdash R_i$ ok. Hence, with Lemma 7.3,

$$\Delta, Y\,\texttt{implements}\,I\langle\overline{V}\rangle, Y \vdash [\overline{V/X}]R_i\ \mathsf{ok}$$

Then $\Delta, Y\,\texttt{implements}\,I\langle\overline{V}\rangle, Y \vdash [\overline{V/X}]K'$ ok. By criterion WF-IFACE-2, $Y \notin \mathsf{ftv}(K')$. With (180) and two applications of Lemma 11.20, we get $\Delta \vdash [\overline{V/X}]K'$ ok as required.

End case distinction on the last rule of the derivation of $K \trianglelefteq_{\mathrm{i}} L$. ☐

Lemma 11.22 (Kernel subtyping propagates well-formedness). *If* $\vdash \Delta$ ok *and* $\Delta \vdash T$ ok *and* $\Delta \vdash_{\mathrm{q}}' T \leq U$ *then* $\Delta \vdash U$ ok.

Proof. Straightforward induction on the derivation of $\Delta \vdash_{\mathrm{q}}' T \leq U$, making use of Lemma 7.16 and Lemma 11.21. ☐

Lemma 11.23 (sup propagates well-formedness). *If* $\Delta \vdash \mathcal{R}$ ok *and* $\mathcal{S} \in \mathsf{sup}(\mathcal{R})$ *then* $\Delta \vdash \mathcal{S}$ ok.

Proof. We proceed by induction on the derivation of $\mathcal{S} \in \mathsf{sup}(\mathcal{R})$. If this derivation ends with rule SUP-ID, then the claim holds trivially. Otherwise, we have

$$\frac{\texttt{interface}\, I\langle\overline{X}\rangle\,[\overline{Y\,\texttt{where}\,\overline{S}}]\,\texttt{where}\,\overline{P}\ \ldots \qquad \overline{U}\,\texttt{implements}\,I\langle\overline{V}\rangle \in \mathsf{sup}(\mathcal{R})}{\underbrace{[\overline{V/X}, \overline{U/Y}]S_j}_{=\mathcal{S}} \in \mathsf{sup}(\mathcal{R})}$$

By the I.H., we have $\Delta \vdash \overline{U}\,\texttt{implements}\,I\langle\overline{V}\rangle$ ok. This derivation must end with rule OK-IMPL-CONSTR. Inverting the rule then yields

$$\Delta \Vdash [\overline{V/X}, \overline{U/Y}]\overline{S}, \overline{P}$$
$$\Delta \vdash \overline{U}, \overline{V}\ \mathsf{ok}$$

The underlying program is well-typed, so we have $\overline{S}, \overline{P}, \overline{X}, \overline{Y} \vdash S_j$ ok. With Lemma 7.3 then $\Delta \vdash \mathcal{S}$ ok. ☐

Lemma 11.24 (Quasi-algorithmic entailment propagates well-formedness)**.** *Assume* $\vdash \Delta\ \mathsf{ok}$ *and* $\Delta \vdash \overline{T}\ \mathsf{ok}$. *If* $\Delta \Vdash_{\mathrm{q}} \overline{T}\ \mathtt{implements}\ I\langle\overline{V}\rangle$ *then* $\Delta \vdash \overline{T}\ \mathtt{implements}\ I\langle\overline{V}\rangle\ \mathsf{ok}$.

Proof. We have

$$\textsc{ent-q-alg-up}\ \frac{\begin{array}{c}(\forall i)\ \Delta \vdash_{\mathrm{q}}{}'\ T_i \leq U_i \\ (\forall i)\ \text{if } T_i \neq U_i \text{ then } i \in \mathsf{pos}^{-}(I) \qquad \Delta \Vdash_{\mathrm{q}}{}'\ \overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle\end{array}}{\Delta \Vdash_{\mathrm{q}} \overline{T}\ \mathtt{implements}\ I\langle\overline{V}\rangle}$$

By Lemma 11.22

$$\Delta \vdash \overline{U}\ \mathsf{ok} \tag{181}$$ {eq:u-ok::lemma:quasi-entail-ok}

Case distinction on the last rule of the derivation of $\Delta \Vdash_{\mathrm{q}}{}'\ \overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle$.

- *Case* rule ent-q-alg-env: Then $R \in \Delta$ and $\overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(R)$. With $\vdash \Delta\ \mathsf{ok}$ we have $\Delta \vdash R\ \mathsf{ok}$. By Lemma 11.23 we have $\Delta \vdash \overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle\ \mathsf{ok}$.

- *Case* rule ent-q-alg-impl: Then

$$\frac{\mathtt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\,\overline{N}\,]\ \mathtt{where}\ \overline{P}\ \ldots \qquad \Delta \Vdash_{\mathrm{q}} [\overline{W/X}]\overline{P}}{\Delta \Vdash_{\mathrm{q}} \underbrace{[\overline{W/X}](\overline{N}\ \mathtt{implements}\ I\langle\overline{V'}\rangle)}_{=\overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle}}$$

  Because the underlying program is well-typed, we have $\overline{P}, \overline{X} \vdash \overline{N}\ \mathtt{implements}\ I\langle\overline{V'}\rangle\ \mathsf{ok}$. Moreover, with (181) $\Delta \vdash [\overline{W/X}]\overline{N}\ \mathsf{ok}$ and by criterion Wf-Impl-1 $\overline{X} \subseteq \mathsf{ftv}(\overline{N})$. Hence, with Lemma 7.38, $\Delta \vdash \overline{W}\ \mathsf{ok}$. Thus, with Lemma 7.3, $\Delta \vdash [\overline{W/X}](\overline{N}\ \mathtt{implements}\ I\langle\overline{V'}\rangle)\ \mathsf{ok}$.

- *Case* rule ent-q-alg-iface: Then

$$\textsc{ent-q-alg-iface}\ \frac{1 \in \mathsf{pos}^{+}(J) \qquad J\langle\overline{W}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{V}\rangle}{\Delta \Vdash_{\mathrm{q}}{}'\ \underbrace{J\langle\overline{W}\rangle\ \mathtt{implements}\ I\langle\overline{V}\rangle}_{=\overline{U}\ \mathtt{implements}\ I\langle\overline{V}\rangle}}$$

  From $\Delta \vdash J\langle\overline{W}\rangle\ \mathsf{ok}$ and Lemma 11.21 we have

$$\Delta \vdash I\langle\overline{V}\rangle\ \mathsf{ok} \tag{182}$$ {eq:1::lemma:quasi-entail-ok}

  Assume

$$\mathtt{interface}\ I\langle\overline{X}\rangle\,[Y\ \mathtt{where}\ \overline{R}]\ \mathtt{where}\ \overline{P}\ldots \tag{183}$$ {eq:2::lemma:quasi-entail-ok}

  From (182) then

$$\begin{gathered}\Delta, Y\ \mathtt{implements}\ I\langle\overline{V}\rangle, Y \Vdash [\overline{V/X}]\overline{R}, \overline{P} \\ Y \notin \mathsf{ftv}(\Delta, \overline{V})\end{gathered} \tag{184}$$ {eq:3::lemma:quasi-entail-ok}

  With $\Delta \Vdash_{\mathrm{q}}{}'\ J\langle\overline{W}\rangle\ \mathtt{implements}\ I\langle\overline{V}\rangle$ we also have $\Delta \Vdash J\langle\overline{W}\rangle\ \mathtt{implements}\ I\langle\overline{V}\rangle$ by Corollary 11.2. Hence,

$$\Delta \Vdash [J\langle\overline{W}\rangle/Y](\Delta, Y\ \mathtt{implements}\ I\langle\overline{V}\rangle, Y)$$

  Thus, with Corollary 6.29 applied on (184)

$$\Delta \Vdash \underbrace{[J\langle\overline{W}\rangle/Y][\overline{V/X}]\overline{R}, \overline{P}}_{=[J\langle\overline{W}\rangle/Y,\overline{V/X}]\overline{R},\overline{P}} \tag{185}$$ {eq:4::lemma:quasi-entail-ok}

  We then have with $\Delta \vdash J\langle\overline{W}\rangle\ \mathsf{ok}$, (182), (183), (185), and rule ok-impl-constr that

$$\Delta \vdash J\langle\overline{W}\rangle\ \mathtt{implements}\ I\langle\overline{V}\rangle\ \mathsf{ok}$$

End case distinction on the last rule of the derivation of $\Delta \Vdash_{\text{q}}{}' \overline{U}\,\texttt{implements}\,I\langle\overline{V}\rangle$. Let

$$\texttt{interface}\ I\langle\overline{X}\rangle\,[\overline{Y}\ \texttt{where}\ \overline{R}]\ \texttt{where}\ \overline{P}\ldots$$

Because we just proved that $\Delta \vdash \overline{U}\,\texttt{implements}\,I\langle\overline{V}\rangle\ \mathsf{ok}$, we have

$$\Delta \vdash \overline{V}\ \mathsf{ok}$$
$$\Delta \Vdash [\overline{V/X}, \overline{U/Y}]\overline{R}, \overline{P}$$

We now prove by induction on the number of indices i with $T_i \neq U_i$ that $\Delta \Vdash [\overline{V/X}, \overline{T/Y}]\overline{R}, \overline{P}$. The original claim then follows with rule OK-IMPL-CONSTR.

- Assume there are no indices i with $T_i \neq U_i$. Then $\Delta \Vdash [\overline{V/X}, \overline{T/Y}]\overline{R}, \overline{P}$ holds trivially.
- Assume i such that $T_i \neq U_i$. The I.H. then gives us that $\Delta \Vdash [\overline{V/X}, \overline{T'/Y}]\overline{R}, \overline{P}$ where

$$T'_j = \begin{cases} T_j & \text{if } i \neq j \\ U_j & \text{if } i = j \end{cases}$$

From $T_i \neq U_i$ we have $i \in \mathsf{pos}^-(I)$. Hence $Y_i \notin \mathsf{ftv}(\overline{P})$. Thus,

$$\Delta \Vdash [\overline{W/X}, \overline{T/Y}]\overline{P}$$

Now suppose $Y_i \in \mathsf{ftv}(\overline{G}\,\texttt{implements}\,J'\langle\overline{W'}\rangle)$ for some $\overline{G}\,\texttt{implements}\,J'\langle\overline{W'}\rangle \in \overline{R}$. Then we have with $i \in \mathsf{pos}^-(I)$ and well-formedness criteria WF-IFACE-2 and WF-IFACE-3 that $Y_i \notin \mathsf{ftv}(\overline{W'})$ and that $Y_i \in \mathsf{ftv}(G_j)$ implies $Y_i = G_j$ and $j \in \mathsf{pos}^-(J')$. Hence, with $\Delta \vdash_{\text{a}}{}' T_i \leq U_i$ and (possibly) some applications of rule ENT-UP, we also get $\Delta \Vdash [\overline{W/X}, \overline{T/Y}]\overline{R}$, as required. □

Lemma 11.25. *If* $\vdash \Delta\ \mathsf{ok}$ *and* $\Delta \vdash T\ \mathsf{ok}$ *and* $\mathsf{bound}_\Delta(T) = N$, *then* $\Delta \vdash N\ \mathsf{ok}$.

PROOF. Follows easily by case distinction on the rule used to derive $\mathsf{bound}_\Delta(T) = N$. We use Lemma 11.22 for the case BOUND-VAR. □

Lemma 11.26. *If* $\Delta \Vdash^?_{\text{a}} \overline{T^?}\,\texttt{implements}\,I\langle\overline{U^?}\rangle \rightharpoonup \overline{T}\,\texttt{implements}\,I\langle\overline{U}\rangle$ *and* $T^?_i \neq \mathsf{nil}$ *then* $T^?_i = T_i$.

PROOF. Follows by straightforward induction on the given derivation. □

Lemma 11.27. *If* $\Delta \vdash_{\text{a}}{}' X \leq I\langle\overline{T}\rangle$ *then* $1 \in \mathsf{pos}^-(I)$.

PROOF. With $\Delta \vdash_{\text{a}}{}' X \leq I\langle\overline{T}\rangle$ also $\Delta \vdash_{\text{q}}{}' X \leq I\langle\overline{T}\rangle$. We then proceed by induction on the derivation of $\Delta \vdash_{\text{q}}{}' X \leq I\langle\overline{T}\rangle$. The derivation must end with an application of rule SUB-Q-ALG-VAR. Hence, $X\ \texttt{extends}\ T \in \Delta$ and $\Delta \vdash_{\text{q}}{}' T \leq I\langle\overline{T}\rangle$.
Case distinction on the form of T.

- *Case* $T = Y$: The claim then follows from the I.H.
- *Case* $T = N$: Impossible by Lemma 6.11.
- *Case* $T = J\langle\overline{U}\rangle$: Then $J\langle\overline{U}\rangle \trianglelefteq_{\text{i}} I\langle\overline{T}\rangle$ by Lemma 6.11 and $1 \in \mathsf{pos}^-(J)$ by criterion WF-TENV-6. The claim now follows with Lemma 6.20.

End case distinction on the form of T. □

Definition 11.28 ($\vdash \Delta\ \mathsf{ok}$ and $\Delta \vdash \Gamma\ \mathsf{ok}$)**.** *A type environment* Δ *is well-formed, written* $\vdash \Delta\ \mathsf{ok}$, *if and only if* $\Delta \vdash P\ \mathsf{ok}$ *for all* $P \in \Delta$. *A value environment* Γ *is well-formed under the type environment* Δ, *written* $\Delta \vdash \Gamma\ \mathsf{ok}$, *if and only if* $\Delta \vdash T :\ \mathsf{ok}$ *for all* $x : T \in \Gamma$.

Lemma 11.29. *Assume* $\mathsf{mtype}_\Delta(m^c, C\langle\overline{W}\rangle) = \langle\overline{X}\rangle\,\overline{U\,x}^n \to U$ `where` $\overline{\mathcal{P}}$ *and let* σ *be a substitution with* $\mathsf{dom}(\sigma) = \overline{X}$*. Suppose* $\vdash \Delta$ ok *and* $\Delta \vdash N$ ok*. If* $N \trianglelefteq_c C\langle\overline{W}\rangle$ *and* $\Delta \Vdash \sigma\overline{\mathcal{P}}$*, then* $\mathsf{a\text{-}mtype}^c(m, N) = \langle\overline{X}\rangle\,\overline{U\,x}^n \to U'$ `where` $\overline{\mathcal{P}}$ *such that* $\Delta \vdash \sigma U' \leq \sigma U$*.*

Proof. From $\mathsf{mtype}_\Delta(m^c, C\langle\overline{W}\rangle) = \langle\overline{X}\rangle\,\overline{U\,x}^n \to U$ `where` $\overline{\mathcal{P}}$ we get

$$
\begin{aligned}
&\texttt{class } C\langle\overline{Y}\rangle \texttt{ extends } M \texttt{ where } \overline{Q}\,\{\ldots\ \overline{m : \mathit{msig}\,\{e\}}\,\}\\
&m_j = m^c\\
&\langle\overline{X}\rangle\,\overline{U\,x}^n \to U \texttt{ where } \overline{\mathcal{P}} = [\overline{W/Y}]\mathit{msig}_j \qquad (186)\ \{\texttt{eq:def-msig-j::lemma:mtype-class-complete}\}
\end{aligned}
$$

Case distinction on the last rule in the derivation of $N \trianglelefteq_c C\langle\overline{W}\rangle$.

- *Case* ext-c-refl: Then $N = C\langle\overline{W}\rangle$, so the claim follows with rule alg-mtype-direct and reflexivity of subtyping.
- *Case* ext-c-super: Then $N = D\langle\overline{V}\rangle$ and

$$
\frac{\texttt{class } D\langle\overline{Z}\rangle \texttt{ extends } M' \texttt{ where } \overline{Q'}\,\{\ldots\ \overline{m' : \mathit{msig}'\,\{e'\}}\,\} \qquad [\overline{V/Z}]M' \trianglelefteq_c C\langle\overline{W}\rangle}{D\langle\overline{V}\rangle \trianglelefteq_c C\langle\overline{W}\rangle}
$$

  Clearly, $D\langle\overline{V}\rangle \trianglelefteq_c [\overline{V/Z}]M'$, so we get with $\Delta \vdash N$ ok and Lemma 7.16 that $\Delta \vdash [\overline{V/Z}]M'$ ok.

  Case distinction on whether or not $m \in \overline{m'}$.

  - *Case* $m \notin \overline{m'}$: The claim then follows from the I.H. and rule alg-mtype-super.
  - *Case* $m \in \overline{m'}$: Assume $m = m'_i$. Because the underlying program is well-typed, we have

$$\overline{Q'}, \overline{Z} \vdash m'_i : \mathit{msig}'_i\,\{e'_i\}\ \mathsf{ok\,in}\ D\langle\overline{Z}\rangle$$

    Hence,

$$\mathsf{override\text{-}ok}_{\overline{Q'},\overline{Z}}(m'_i : \mathit{msig}'_i, D\langle\overline{Z}\rangle)$$

    With $D\langle\overline{V}\rangle \trianglelefteq_c C\langle\overline{W}\rangle$ and Lemma 7.14 there exists $\overline{W'}$ such that

$$
\begin{aligned}
&D\langle\overline{Z}\rangle \trianglelefteq_c C\langle\overline{W'}\rangle\\
&[\overline{V/Z}]\overline{W'} = \overline{W} \qquad (187)\ \{\texttt{eq:subst-vz::lemma:mtype-class-complete}\}
\end{aligned}
$$

    By inverting rule ok-override

$$\overline{Q'}, \overline{Z} \vdash \mathit{msig}'_i \leq [\overline{W'/Y}]\mathit{msig}_j$$

    Assume

$$
\begin{aligned}
\mathit{msig}'_i &= \langle\overline{X'''}\rangle\,\overline{U'''\,x'''} \to U''' \texttt{ where } \overline{P'''}\\
\mathit{msig}_j &= \langle\overline{X''}\rangle\,\overline{U''\,x''} \to U'' \texttt{ where } \overline{P''}
\end{aligned}
$$

    Then by rule sub-msig

$$
\begin{aligned}
&\overline{X'''} = \overline{X''}\\
&\overline{U'''} = [\overline{W'/Y}]\overline{U''}\\
&\overline{x'''} = \overline{x''}\\
&\overline{P'''} = [\overline{W'/Y}]\overline{P''} \qquad (188)\ \{\texttt{eq:p'''::lemma:mtype-class-complete}\}\\
&\overline{Q'}, \overline{Z}, \overline{P'''}, \overline{X'''} \vdash U''' \leq [\overline{W'/Y}]U'' \qquad (189)\ \{\texttt{eq:sub::lemma:mtype-class-complete}\}
\end{aligned}
$$

From (186)

$$\overline{X''} = \overline{X}$$
$$[\overline{W/Y}]\overline{U''} = \overline{U}$$
$$\overline{x''} = \overline{x}$$
$$[\overline{W/Y}]U'' = U$$
$$[\overline{W/Y}]\overline{P''} = \overline{\mathcal{P}}$$

Moreover, we have with (187) and the fact that $\overline{Z} \cap \mathsf{ftv}(\overline{U''}, U'', \overline{P''}) = \emptyset$

$$[\overline{V/Z}][\overline{W'/Y}](\overline{U''}, U'', \overline{P''}) = [\overline{W/Y}](\overline{U''}, U'', \overline{P''}) = (\overline{U}, U, \overline{\mathcal{P}}) \tag{190}$$ {eq:eqs::lemma:mtype-class-complete}

Hence, we have with rule ALG-MTYPE-DIRECT

$$\begin{aligned}
\mathsf{a\text{-}mtype}^{\mathrm{c}}(m, D\langle\overline{V}\rangle) &= [\overline{V/Z}]\mathit{msig}'_i \\
&= [\overline{V/Z}](\langle\overline{X'''}\rangle\, \overline{U'''\, x'''} \to U''' \ \texttt{where}\ \overline{P'''}) \\
&= [\overline{V/Z}](\langle\overline{X}\rangle\, \overline{[\overline{W'/Y}]U''\, x} \to U''' \ \texttt{where}\ \overline{[\overline{W'/Y}]P''}) \\
&= \langle\overline{X}\rangle\, \overline{[\overline{W/Y}]U''\, x} \to [\overline{V/Z}]\overline{U'''} \ \texttt{where}\ [\overline{W/Y}]\overline{P''} \\
&= \langle\overline{X}\rangle\, \overline{U\, x} \to [\overline{V/Z}]\overline{U'''} \ \texttt{where}\ \overline{\mathcal{P}}
\end{aligned}$$

To finish this case, we still need to show that for $U' = [\overline{V/Z}]\overline{U'''}$ we have $\Delta \vdash \sigma U' \leq \sigma U$. From the assumption $\Delta \vdash D\langle\overline{V}\rangle$ ok we get $\Delta \Vdash [\overline{V/Z}]\overline{Q'}$. W.l.o.g. $\overline{X} \cap \mathsf{ftv}([\overline{V/Z}]\overline{Q'}) = \emptyset$. Hence, $\Delta \Vdash \sigma[\overline{V/Z}]\overline{Q'}$. From (188) and (190) and the assumption $\Delta \Vdash \sigma\overline{\mathcal{P}}$ we get $\Delta \Vdash \sigma[\overline{V/Z}]\overline{P'''}$. Thus, with (189) and Corollary 6.29

$$\Delta \vdash \sigma[\overline{V/Z}]U''' \leq \sigma[\overline{V/Z}][\overline{W'/Y}]U''$$

But with (190) we have $\sigma[\overline{V/Z}][\overline{W'/Y}]U'' = \sigma U$.

End case distinction on whether or not $m \in \overline{m'}$.

End case distinction on the last rule in the derivation of $N \trianglelefteq_{\mathrm{c}} C\langle\overline{W}\rangle$. □

Lemma 11.30. *Assume* $\mathsf{mtype}_\Delta(m, T) = \langle\overline{X}\rangle\, \overline{U\, x}^n \to U$ $\texttt{where}$ $\overline{\mathcal{P}}$ *and let* σ *be a substitution with* $\mathsf{dom}(\sigma) = \overline{X}$. *Suppose* $\vdash \Delta$ ok *and* $\Delta \vdash T'$ ok. *If* $\Delta \vdash T' \leq T$, $\Delta \vdash T_i \leq \sigma U_i$ *for all* $i \in [n]$, *and* $\Delta \Vdash \sigma\overline{\mathcal{P}}$, *then* $\mathsf{a\text{-}mtype}_\Delta(m, T', \overline{T}) = \langle\overline{X}\rangle\, \overline{U'\, x}^n \to U'$ $\texttt{where}$ $\overline{\mathcal{P}}$ *such that* $\Delta \vdash T_i \leq \sigma U'_i$ *for all* $i \in [n]$ *and* $\Delta \vdash \sigma U' \leq \sigma U$.

PROOF. *Case distinction* on the form of m.

- *Case* $m = m^{\mathrm{c}}$: Then $T = C\langle\overline{W}\rangle$. We have by Lemma 6.16 that $\Delta \vdash_{\mathrm{a}}{}' T' \leq C\langle\overline{W}\rangle$. By Lemma 11.18 we have

$$\mathsf{bound}_\Delta(T') = N$$
$$N \trianglelefteq_{\mathrm{c}} C\langle\overline{W}\rangle$$

With Lemma 11.25 we get $\Delta \vdash N$ ok. The claim now follows with Lemma 11.29.

- *Case* $m = m^{\mathrm{i}}$: From $\mathsf{mtype}_\Delta(m, T) = \langle \overline{X} \rangle\, \overline{U\, x}^n \to U$ `where` $\overline{\mathcal{P}}$ we get

$$
\begin{aligned}
\texttt{interface}\ I\langle \overline{Z'} \rangle\, [\overline{Z}^l\ \texttt{where}\, \overline{R}]\ \texttt{where}\ \overline{P}\, \{\ldots\ \overline{rcsig}\,\} \\
rcsig_j = \texttt{receiver}\, \{\overline{m : msig}\} \\
m = m_k \\
msig_k = \langle \overline{X} \rangle\, \overline{U''\, x} \to U''\ \texttt{where}\ \overline{P''} \\
\Delta \Vdash \overline{T'}\ \texttt{implements}\, I\langle \overline{W} \rangle & \qquad (191) \quad \{\texttt{eq:entails-mt'::lemma:mtype-complete}\} \\
T'_j = T \\
(\overline{U}, U, \overline{\mathcal{P}}) = [\overline{T'/Z}, \overline{W/Z'}](\overline{U''}, U'', \overline{P''}) & \qquad (192) \quad \{\texttt{eq:eqs-up::lemma:mtype-complete}\}
\end{aligned}
$$

By Lemma 6.32, there are two possibilities.

Case distinction on the possibilities left by Lemma 6.32.

  - *Case* 1st possibility:

$$
\begin{aligned}
[l] = \mathscr{N}_1 \,\dot\cup\, \mathscr{N}_2 \\
T'_i = K_i \text{ for all } i \in \mathscr{N}_1 \\
i \in \mathsf{pos}^-(I) \text{ for all } i \in \mathscr{N}_1 & \qquad (193) \quad \{\texttt{eq:n1-neg::lemma:mtype-complete}\} \\
T'_i = G_i \text{ for all } i \in \mathscr{N}_2 & \qquad (194) \quad \{\texttt{eq:n2-gtype::lemma:mtype-complete}\} \\
\Delta \Vdash \overline{T''}\ \texttt{implements}\, I\langle \overline{W} \rangle \text{ for all } \overline{T''} \text{ with } T''_i = G_i \text{ for all } i \in \mathscr{N}_2 & \qquad (195) \quad \{\texttt{eq:impl-for-all::lemma:mtype-complete}\}
\end{aligned}
$$

    Define for all $i \in [l]$:

$$
\mathscr{V}_i^? = \begin{cases} \mathsf{contrib}'_{\Delta;Z_i}(\overline{U''}, \overline{T}) & \text{if } i \neq j \\ \mathsf{contrib}'_{\Delta;Z_i}(Z_j\, \overline{U''}, T'\, \overline{T}) & \text{if } i = j \end{cases} \qquad (196) \quad \{\texttt{eq:def-set-vi::lemma:mtype-complete}\}
$$

$$
V_i^? = \begin{cases} \mathsf{nil} & \text{if } \mathscr{V}_i^? = \mathsf{nil} \\ T'_i & \text{if } \mathscr{V}_i^? \neq \mathsf{nil} \text{ and } i \in \mathscr{N}_2 \\ \texttt{Object} & \text{if } \mathscr{V}_i^? \neq \mathsf{nil} \text{ and } i \in \mathscr{N}_1 \end{cases} \qquad (197) \quad \{\texttt{eq:def-vi::lemma:mtype-complete}\}
$$

    We now prove

$$
\begin{aligned}
\textit{for all } i \in [l],\ \textit{either } V_i^? = \mathsf{nil} \\
\textit{or } V_i^? \neq \mathsf{nil} \textit{ and } \Delta \vdash_{\mathrm{a}}{}' V'_i \leq V_i^? \textit{ for some } V'_i \in \mathscr{V}_i^? & \qquad (198) \quad \{\texttt{eq:aux::lemma:mtype-complete}\}
\end{aligned}
$$

    Assume $i \in [l]$.
    Case distinction on whether or not $\mathscr{V}_i^? = \mathsf{nil}$.

    * *Case* $\mathscr{V}_i^? = \mathsf{nil}$: Then $V_i^? = \mathsf{nil}$. Thus, (198) holds for this specific i.
    * *Case* $\mathscr{V}_i^? \neq \mathsf{nil}$: Define

$$
\mathscr{T}_i = \{T_q \mid q \in [n], U''_q = Z_i\} \cup (\text{if } i = j \text{ then } \{T'\} \text{ else } \emptyset)
$$

      Then

$$
\mathscr{V}_i^? = \mathsf{MUB}_\Delta \mathscr{T}_i \neq \mathsf{nil} \qquad (199) \quad \{\texttt{eq:vi-mub::lemma:mtype-complete}\}
$$

      by definition of $\mathsf{contrib}'$. With Corollary 11.15 we get $\mathscr{V}_i^? \neq \emptyset$.
      If $i \in \mathscr{N}_1$ then $V_i^? = \texttt{Object}$, so (198) holds for this specific i.
      Now suppose $i \in \mathscr{N}_2$. Then $T'_i = G_i$ by (194). From the assumptions we get

$$
(\forall q \in [n])\ \Delta \vdash T_q \leq \sigma U_q
$$

Let $q \in [n]$ such that $U''_q = Z_i$. W.l.o.g., $\overline{X} \cap \mathsf{ftv}(\overline{T'}) = \emptyset$. Hence, with (192)

$$\sigma U_q = \sigma T'_i = T'_i = G_i$$

Thus, with Lemma 6.16

$$\Delta \vdash_{\mathrm{a}}{}' T_q \leq T'_i$$

If $i = j$ then we also have $T'_i = T'_j = T$, so by the assumption $\Delta \vdash T' \leq T$

$$\Delta \vdash T' \leq T'_i$$

Then again with Lemma 6.16

$$\Delta \vdash_{\mathrm{a}}{}' T' \leq T'_i$$

Hence,

$$\Delta \vdash_{\mathrm{a}}{}' \tilde{T} \leq T'_i \text{ for all } \tilde{T} \in \mathscr{T}_i$$

By (199) and Lemma 11.14, there exists $V'_i \in \mathscr{V}_i^?$ such that

$$\Delta \vdash_{\mathrm{a}}{}' V'_i \leq T'_i$$

But $V_i^? = T'_i$ because $i \in \mathscr{N}_2$.

End case distinction on whether or not $\mathscr{V}_i^? = \mathsf{nil}$.
This finishes the proof of (198).
Now define

$$\begin{aligned} \mathscr{M} = \{\overline{V}\,\texttt{implements}\,I\langle\overline{V''}\rangle \mid (\forall i \in [l])\ & \text{if } \mathscr{V}_i^? = \mathsf{nil} \text{ then } V_i^? = \mathsf{nil} \\ & \text{else define } V_i^? \text{ such that} \\ & \quad \Delta \vdash_{\mathrm{a}}{}' V'_i \leq V_i^? \text{ for } V'_i \in \mathscr{V}_i^?, \\ \Delta \Vdash_{\mathrm{a}}^? \overline{V^?}\,\texttt{implements}\,I\langle\overline{\mathsf{nil}}\rangle & \rightharpoonup \overline{V}\,\texttt{implements}\,I\langle\overline{V''}\rangle\} \end{aligned} \tag{200}$$ {eq:def-set-m::lemma:mtype-complete}

We now show that $\mathscr{M} \neq \emptyset$. Define for all $i \in [l]$

$$T'''_i = \begin{cases} T'_i & \text{if } V_i^? = \mathsf{nil} \\ V_i^? & \text{otherwise} \end{cases} \tag{201}$$ {eq:def-t'''::lemma:mtype-complete}

With (195) and the definition of $V_i^?$:

$$\Delta \Vdash \overline{T'''}\,\texttt{implements}\,I\langle\overline{W}\rangle \tag{202}$$ {eq:entails::lemma:mtype-complete}

Clearly, $\overline{V^?}\,\overline{\mathsf{nil}} \sharp \overline{T'''}\,\overline{W}$ and $V_i^? \neq \mathsf{nil}$ if $i \in \mathsf{disp}(I)$. Hence, by Lemma 11.5

$$\Delta \Vdash_{\mathrm{a}}^? \overline{V^?}\,\texttt{implements}\,I\langle\overline{\mathsf{nil}}\rangle \rightharpoonup \overline{W'}\,\texttt{implements}\,I\langle\overline{W}\rangle$$

for $\overline{W'}$ such that

$$T'''_i = W'_i \text{ if } V_i^? \neq \mathsf{nil} \text{ or } i \notin \mathsf{pos}^-(I) \tag{203}$$ {eq:ti'''-eq-wi'::lemma:mtype-complete}

With (198) we thus have

$$\overline{W'}\,\texttt{implements}\,I\langle\overline{W}\rangle \in \mathscr{M} \tag{204}$$ {eq:elem-M::lemma:mtype-complete}

so

$$\mathscr{M} \neq \emptyset \tag{205}$$ {eq:M-not-empty::lemma:mtype-complete}

Moreover, for all $\overline{V}$ **implements** $I\langle\overline{V''}\rangle \in \mathscr{M}$ the following holds:

$$\Delta \vdash_{\mathrm{a}}' T_q \leq V_i \text{ for all } i \in [l], q \in [n] \text{ with } U''_q = Z_i \tag{206}$$ {eq:1::lemma:mtype-complete}

$$\Delta \vdash_{\mathrm{a}}' T' \leq V_j \tag{207}$$ {eq:2::lemma:mtype-complete}

$$\overline{V''} = \overline{W} \tag{208}$$ {eq:3::lemma:mtype-complete}

$$V_i = T'_i \text{ for all } i \in [l], i \notin \mathsf{disp}(I) \cup \mathsf{pos}^-(I) \tag{209}$$ {eq:4::lemma:mtype-complete}

* Equations (206) and (207) follow from (200) and (196) and with Lemma 11.26.
* To prove equations (208) and (209), proceed as follows: We have by Lemma 11.3 and (200) that

  $$\Delta \Vdash \overline{V}\ \mathtt{implements}\ I\langle\overline{V''}\rangle$$

  With (202) we get

  $$\Delta \Vdash \overline{T'''}\ \mathtt{implements}\ I\langle\overline{W}\rangle$$

  Suppose $i' \in \mathsf{disp}(I)$. Clearly, $\mathscr{V}^?_{i'} \neq \mathsf{nil}$. Thus, using Lemma 11.26, (204), (203), and (200) there exists $V'_{i'}, V''_{i'} \in \mathscr{V}^?_{i'}$ such that

  $$\Delta \vdash_{\mathrm{a}}' V'_{i'} \leq V_{i'}$$
  $$\Delta \vdash_{\mathrm{a}}' V''_{i'} \leq T'''_{i'}$$

  Define $T'' = T'$ if $i' = j$ and $T'' = T_q$ for some $q \in [n]$ with $U''_q = Z_{i'}$ otherwise. By (196), the definition of $\mathsf{contrib}'$, and Lemma 11.13 we have

  $$\Delta \vdash_{\mathrm{a}}' T'' \leq V'_{i'}$$
  $$\Delta \vdash_{\mathrm{a}}' T'' \leq V''_{i'}$$

  Hence,

  $$\Delta \vdash_{\mathrm{a}}' T'' \leq V_{i'}$$
  $$\Delta \vdash_{\mathrm{a}}' T'' \leq T'''_{i'}$$

  With Lemma 11.11 we then get

  $$\overline{V''} = \overline{W}$$
  $$V_i = T'''_i \text{ for all } i \notin \mathsf{disp}(I) \cup \mathsf{pos}^-(I)$$

  This proves (208). Now assume $i \notin \mathsf{disp}(I) \cup \mathsf{pos}^-(I)$. Then $i \in \mathscr{N}_2$ by (193). By (201) and (197) we have $T'''_i = T'_i$. This proves (209).

Define

$$p^? = \begin{cases} i & \text{if } U'' = Z_i \\ \mathsf{nil} & \text{otherwise} \end{cases} \tag{210}$$ {eq:def-p::lemma:mtype-complete}

Now assume

$$\mathsf{pick\text{-}constr}^{p^?}_{\Delta} \mathscr{M} = \overline{V}\ \mathtt{implements}\ I\langle\overline{V''}\rangle \tag{211}$$ {eq:assum::lemma:mtype-complete}

for some $\overline{V}$ **implements** $I\langle\overline{V''}\rangle$. (We will prove (211) shortly.)
We then can use rule ALG-MTYPE-IFACE to derive

$$\begin{aligned}\mathsf{a\text{-}mtype}_{\Delta}(m, T', \overline{T}) &= [\overline{V/Z}, \overline{V''/Z'}]\mathit{msig}_k \\ &= [\overline{V/Z}, \overline{V''/Z'}](\langle\overline{X}\rangle\, \overline{U''\,x} \to U''\ \mathtt{where}\ \overline{P''})\end{aligned}$$

From criterion Wf-Iface-4 we have $\overline{Z} \cap \mathsf{ftv}(\overline{P'''}) = \emptyset$. With (192) and (208) we thus get

$$[\overline{V/Z}, \overline{V''/Z'}]\overline{P''} = \overline{\mathcal{P}}$$

Now suppose $i \in [n]$. Define $U'_i = [\overline{V/Z}, \overline{V''/Z'}]U''_i$.

* If $\overline{Z} \cap \mathsf{ftv}(U''_i) = \emptyset$ then with (192) and (208)

  $$\sigma U'_i = \sigma[\overline{V/Z}, \overline{V''/Z'}]U''_i = \sigma[\overline{V''/Z'}]U''_i = \sigma U_i$$

  We now get

  $$\Delta \vdash T_i \leq \sigma U'_i$$

  by the assumption $\Delta \vdash T_i \leq \sigma U_i$.
* If $\overline{Z} \cap \mathsf{ftv}(U''_i) \neq \emptyset$ then by criterion Wf-Iface-4 $U''_i = Z_{i'}$ for some $i' \in [l]$. W.l.o.g., $\overline{X} \cap \mathsf{ftv}(\overline{V}) = \emptyset$. Hence,

  $$\sigma U'_i = \sigma[\overline{V/Z}, \overline{V''/Z'}]U''_i = \sigma V_{i'} = V_{i'}$$

  With (206) we have

  $$\Delta \vdash_{\mathrm{a}}{}' T_i \leq V_{i'}$$

  Hence,

  $$\Delta \vdash T_i \leq \sigma U'_i$$

Thus, $\Delta \vdash T_i \leq \sigma U'_i$ for all $i \in [n]$.
Define

$$U' = [\overline{V/Z}, \overline{V''/Z'}]U'' \qquad (212)$$ {eq:def-u'::lemma:mtype-complete}

We still need to prove $\Delta \vdash \sigma U' \leq \sigma U$ and (211).
Case distinction on whether or not $U'' \in \overline{Z}$.

* *Case* $U'' \notin \overline{Z}$: Then $\overline{Z} \cap \mathsf{ftv}(U'') = \emptyset$ by criterion Wf-Iface-4. By (210) we have $p^? = \mathsf{nil}$. Then (211) holds trivially by rule pick-constr-nil. Moreover, we have with (192) and (208) that

  $$\sigma U' = \sigma[\overline{V/Z}, \overline{V''/Z'}]U'' = \sigma[\overline{V''/Z'}]U'' = \sigma U$$

* *Case* $U'' \in \overline{Z}$: Then $U'' = Z_i$ for some $i \in [l]$ by criterion Wf-Iface-4. By (210) we have $p^? = i$. Moreover,

  $$i \notin \mathsf{pos}^-(I) \qquad (213)$$ {eq:i-not-neg::lemma:mtype-complete}

  In the following, we use the notation $\mathsf{impl}(\mathcal{R}, q)$ to denote the q-th implementing type of $\mathcal{R}$; that is, $\mathsf{impl}(\overline{T}\ \mathtt{implements}\ K, q) := T_q$.
  Case distinction on whether or not $V^?_i = \mathsf{nil}$.
  · *Case* $V^?_i = \mathsf{nil}$: By (197) $\mathscr{V}^?_i = \mathsf{nil}$, so we get by (196) and the definition of $\mathsf{contrib}'$ that $Z_i \notin \overline{U''}$ and $i \neq j$. Thus, it is easy to verify that $i \notin \mathsf{disp}(I)$. With (213) then $i \notin \mathsf{disp}(I) \cup \mathsf{pos}^-(I)$. Hence, for all $\mathcal{R} \in \mathscr{M}$, $\mathsf{impl}(\mathcal{R}, i) = T'_i$ by (209). By rule pick-constr-non-nil we get (211). Obviously, $\overline{V}\ \mathtt{implements}\ I\langle\overline{V''}\rangle \in \mathscr{M}$, so $V_i = T'_i$. With (192), (212), and the fact $U'' = Z_i$ then

  $$\sigma U' = \sigma[\overline{V/Z}, \overline{V''/Z'}]U'' = \sigma V_i = \sigma T'_i = \sigma U$$

· *Case* $V_i^? \neq \mathsf{nil}$: Because of (213) we have by (193) and (197)

$$i \in \mathscr{N}_2 \tag{214}$$ {eq:i-in-n2::lemma:mtype-complete}

$$V_i^? = T_i'$$

$$\mathscr{V}_i^? \neq \mathsf{nil} \tag{215}$$ {eq:set-vi-not-nil::lemma:mtype-complete}

Suppose $\mathcal{R} \in \mathscr{M}$. By (200) and (208)

$$\mathcal{R} = \dots \texttt{ implements } I\langle\overline{W}\rangle$$

With (200), Lemma 11.3, and Lemma 11.26

$$\Delta \Vdash \mathcal{R}$$

$$\Delta \vdash_{\mathrm{a}}{}' V_{i,\mathcal{R}} \leq \mathsf{impl}(\mathcal{R}, i) \text{ for some } V_{i,\mathcal{R}} \in \mathscr{V}_i^? \tag{216}$$ {eq:vi-sub-impl-R::lemma:mtype-complete}

Next, we show that

$$\mathsf{impl}(\mathcal{R}, i) = G_{i,\mathcal{R}} \tag{217}$$ {eq:impl-i-gtype::lemma:mtype-complete}

for some $G_{i,\mathcal{R}}$. Assume that this is not the case; that is, $\mathsf{impl}(\mathcal{R}, i)$ is an interface type. Because of (213) we get by Lemma 6.32

$$[l] = \{1\}$$

$$\mathcal{R} = J\langle\overline{W''}\rangle \texttt{ implements } I\langle\overline{W}\rangle \tag{218}$$ {eq:def-R::lemma:mtype-complete}

$$J\langle\overline{W''}\rangle \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle \tag{219}$$ {eq:jwp-sub-jw::lemma:mtype-complete}

$$1 \in \mathsf{pos}^+(I)$$

$$1 \in \mathsf{pos}^+(J)$$

Hence, $i = j = 1$. Because $1 \in \mathsf{pos}^+(I)$ we have $Z_i \notin \mathsf{ftv}(\overline{U''})$. With (214) and (194) $T' = G$ for some G, so we have with (196) and the definition of $\mathsf{contrib}'$ that $\mathscr{V}_i^? = \{G\}$. With (216) and (218) then

$$\Delta \vdash_{\mathrm{a}}{}' G \leq J\langle\overline{W''}\rangle$$

By Lemma 6.11 we then have $G = X$ for some X. Thus, by Lemma 11.27

$$1 \in \mathsf{pos}^-(J)$$

With (219) and Lemma 6.20 then also $1 \in \mathsf{pos}^-(I)$, which is a contradiction to (213). This finishes the proof of (217).
Our next goal is to prove that there exists some $\mathcal{R}' \in \mathscr{M}$ such that

$$\Delta \vdash_{\mathrm{a}}{}' \mathsf{impl}(\mathcal{R}', i) \leq \mathsf{impl}(\mathcal{R}, i) \tag{220}$$ {eq:sub-R'::lemma:mtype-complete}

for all $\mathcal{R} \in \mathscr{M}$. Together with (205), this allows us to use rule PICK-CONSTR-NON-NIL to derive (211), yielding

$$\mathsf{pick\text{-}constr}_{\Delta}^{p^?} \mathscr{M} = \overline{V} \texttt{ implements } I\langle\overline{V''}\rangle = \mathcal{R}' \tag{221}$$ {eq:def-R'::lemma:mtype-complete}

W.l.o.g., assume that $\mathsf{impl}(\mathcal{R}, i) \neq \texttt{Object}$ for all $\mathcal{R} \in \mathscr{M}$. (If $\mathsf{impl}(\mathcal{R}, i) = \texttt{Object}$ then (220) holds trivially for this $\mathcal{R}$.) Hence, we have with (217)

$$\mathsf{impl}(\mathcal{R}, i) = G_{i,\mathcal{R}} \neq \texttt{Object}$$

With (216), Lemma 6.11, and Lemma 11.8 we then get

$$\mathscr{V}_i^? \ni V_{i,\mathcal{R}} = H_{i,\mathcal{R}} \neq \texttt{Object} \tag{222}$$ {eq:vi-gtype::lemma:mtype-complete}

By (196) and the definition of $\mathsf{contrib}'$

$$\mathscr{V}_i^? = \mathsf{MUB}_\Delta \underbrace{\left(\{T_q \mid q \in [n], U_q'' = Z_i\} \cup (\text{if } i = j \text{ then } \{T'\} \text{ else } \emptyset)\right)}_{=:\mathscr{T}}$$

Hence, because $V_{i,\mathcal{R}} \in \mathscr{V}_i^?$, we have with Lemma 11.13

$$\Delta \vdash_{\mathrm{a}}' T_q \leq V_{i,\mathcal{R}} \text{ for all } q \in [n], U_q'' = Z_i$$
$$\Delta \vdash_{\mathrm{a}}' T' \leq V_{i,\mathcal{R}} \text{ if } i = j$$

By (196), (215), and the definition of $\mathsf{contrib}'$, we get $\mathscr{T} \neq \emptyset$. By (216), (222), and Lemma 11.16, we get that there exists $V_i \in \mathscr{V}_i^?$ such that $V_i = V_{i,\mathcal{R}}$ for all $\mathcal{R} \in \mathscr{M}$. Hence, with (216),

$$\Delta \vdash_{\mathrm{a}}' V_i \leq \mathsf{impl}(\mathcal{R}, i) \tag{223}$$ {eq:one-vi-sub-impl::lemma:mtype-complete}

for all $\mathcal{R} \in \mathscr{M}$. Now suppose $\mathcal{R}_1, \mathcal{R}_2 \in \mathscr{M}$. We then have $\Delta \vdash_{\mathrm{a}}' V_i \leq \mathsf{impl}(\mathcal{R}_1, i)$ and $\Delta \vdash_{\mathrm{a}}' V_i \leq \mathsf{impl}(\mathcal{R}_2, i)$, so with Lemma 11.6 and (217)

$$\Delta \vdash_{\mathrm{a}}' \mathsf{impl}(\mathcal{R}_1, i) \leq \mathsf{impl}(\mathcal{R}_2, i) \text{ or } \Delta \vdash_{\mathrm{a}}' \mathsf{impl}(\mathcal{R}_2, i) \leq \mathsf{impl}(\mathcal{R}_1, i)$$

But with (223) and Lemma 11.12, we know that the set $\{\mathsf{impl}(\mathcal{R}, i) \mid \mathcal{R} \in \mathscr{M}\}$ is finite. Thus, there exists some $\mathcal{R}' \in \mathscr{M}$ such that $\Delta \vdash_{\mathrm{a}}' \mathsf{impl}(\mathcal{R}', i) \leq \mathsf{impl}(\mathcal{R}, i)$. This finishes the proof of (220) and thus the proof of (211).
Finally, we prove $\Delta \vdash \sigma U' \leq \sigma U$. With (204) we have some $\mathcal{R}'' \in \mathscr{M}$ such that

$$\mathsf{impl}(\mathcal{R}'', i) = W_i' \overset{(203),(213)}{=} T_i''' \overset{(201)}{=} V_i^? \overset{(214),(197)}{=} T_i'$$

By (220) then

$$\Delta \vdash_{\mathrm{a}}' \mathsf{impl}(\mathcal{R}', i) \leq T_i' \tag{224}$$ {eq:impl-sub-ti'::lemma:mtype-complete}

We also have (note $U'' = Z_i$)

$$U' \overset{(212)}{=} [\overline{V/Z}, \overline{V''/Z'}]U'' = [\overline{V/Z}, \overline{V''/Z'}]Z_i = V_i \overset{(221)}{=} \mathsf{impl}(\mathcal{R}', i)$$
$$U \overset{(192)}{=} T_i'$$

W.l.o.g., $\overline{X} \cap \mathsf{ftv}(\overline{V}) = \emptyset = \overline{X} \cap \mathsf{ftv}(\overline{T'})$. Thus, with (224), $\Delta \vdash \sigma U' \leq \sigma U$, as required.

End case distinction on whether or not $V_i^? = \mathsf{nil}$.

End case distinction on whether or not $U'' \in \overline{Z}$.

– *Case* 2nd possibility:

$$[l] = \{1\}$$
$$1 \in \mathsf{pos}^+(I)$$
$$T = T_1' = K \tag{225}$$ {eq:t-eq-k::lemma:mtype-complete}
$$K \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle \tag{226}$$ {eq:k-extends-iw::lemma:mtype-complete}

(By abuse of notation, we identify $\mathsf{pos}(K)$ with $\mathsf{pos}(J)$ for $K = J\langle\overline{T}\rangle$.) Because $1 \in \mathsf{pos}^+(I)$ we have

$$Z_1 \notin \mathsf{ftv}(\overline{U''}) \tag{227}$$ {eq:z1-notin-u''::lemma:mtype-complete}

Define

$$\mathscr{V}_1^? = \mathsf{contrib}'_{\Delta;Z_1}(Z_1\,\overline{U''}, T'\,\overline{T}) = \mathsf{MUB}_\Delta\{T'\} = \{T'\}$$
$$p^? = (\text{if } U'' = Z_1 \text{ then } 1 \text{ else } \mathsf{nil})$$
$$\begin{aligned}\mathscr{M} = \{&V\,\texttt{implements}\,I\langle\overline{V''}\rangle \mid V' \in \mathscr{V}_1^?, \Delta \vdash_{\mathrm{a}}{}' V' \leq V,\\ &\quad \Delta \Vdash_{\mathrm{a}}^? V\,\texttt{implements}\,I\langle\overline{\mathsf{nil}}\rangle \twoheadrightarrow V\,\texttt{implements}\,I\langle\overline{V''}\rangle\}\\ \{&V\,\texttt{implements}\,I\langle\overline{V''}\rangle \mid \Delta \vdash_{\mathrm{a}}{}' T' \leq V,\\ &\quad \Delta \Vdash_{\mathrm{a}}^? V\,\texttt{implements}\,I\langle\overline{\mathsf{nil}}\rangle \twoheadrightarrow V\,\texttt{implements}\,I\langle\overline{V''}\rangle\}\end{aligned} \tag{228}$$ {eq:def-set-m2::lemma:mtype-complete}

We now prove that there exists some T'' such that

$$T''\,\texttt{implements}\,I\langle\overline{W}\rangle \in \mathscr{M} \tag{229}$$ {eq:t''-impl-iw::lemma:mtype-complete}
$$\Delta \vdash T'' \leq K \tag{230}$$ {eq:t''-st-K::lemma:mtype-complete}

From the assumption $\Delta \vdash T' \leq T$ and $T = K$ we get $\Delta \vdash_{\mathrm{a}} T' \leq K$.
Case distinction on whether or not $\Delta \vdash_{\mathrm{a}}{}' T' \leq K$.

* *Case* $\Delta \vdash_{\mathrm{a}}{}' T' \leq K$: From (191) we have $\Delta \Vdash_{\mathrm{a}} K\,\texttt{implements}\,I\langle\overline{W}\rangle$, so with Lemma 11.5 we get that $K\,\texttt{implements}\,I\langle\overline{W}\rangle \in \mathscr{M}$. The claims (229) and (230) then follow for $T'' = K$.
* *Case* not $\Delta \vdash_{\mathrm{a}}{}' T' \leq K$: Hence, by inverting rule SUB-Q-ALG-IMPL,

$$\Delta \vdash_{\mathrm{q}}{}' T' \leq T''$$
$$\Delta \Vdash_{\mathrm{q}}{}' T''\,\texttt{implements}\,K \tag{231}$$ {eq:t''-impl-K::lemma:mtype-complete}

  By rule SUB-IMPL then $\Delta \vdash T'' \leq K$. This proves (230). With (226), Lemma 6.26, and Lemma 6.28, we get $\Delta \Vdash_{\mathrm{a}} T''\,\texttt{implements}\,I\langle\overline{W}\rangle$. With Lemma 11.5 and (228) then

$$T''\,\texttt{implements}\,I\langle\overline{W}\rangle \in \mathscr{M}$$

  This proves (229).

End case distinction on whether or not $\Delta \vdash_{\mathrm{a}}{}' T' \leq K$.
This finishes the proof of (229) and (230).
Let $\mathcal{R} \in \mathscr{M}$. By (228) and Lemma 11.3:

$$\Delta \Vdash_{\mathrm{a}} \mathcal{R} \tag{232}$$ {eq:entails-gi::lemma:mtype-complete}
$$\Delta \vdash_{\mathrm{a}}{}' T' \leq \mathsf{impl}(\mathcal{R}, 1) \tag{233}$$ {eq:t'-st-impl::lemma:mtype-complete}

Moreover, we have with Lemma 11.11, (229), and (228) that

$$\mathcal{R} = V_{\mathcal{R}}\,\texttt{implements}\,I\langle\overline{W}\rangle \tag{234}$$ {eq:form-gi::lemma:mtype-complete}

for some $V_{\mathcal{R}}$.
Case distinction on the form of $p^?$.

* *Case* $p^? = \mathsf{nil}$: Then $U'' \neq Z_1$. By criterion WF-IFACE-4

$$\overline{Z} \cap \mathsf{ftv}(U'') = \emptyset$$
$$\overline{Z} \cap \mathsf{ftv}(\overline{P''}) = \emptyset$$

  Moreover, by (229) we know that $\mathscr{M} \neq \emptyset$, so with (234)

$$\mathsf{pick\text{-}constr}_\Delta^{p^?}\mathscr{M} = V\,\texttt{implements}\,I\langle\overline{W}\rangle$$

for some V `implements` $I\langle\overline{W}\rangle \in \mathscr{M}$. We then have by rule ALG-MTYPE-IFACE and (227)

$$\begin{aligned}\mathsf{a\text{-}mtype}_\Delta(m, T', \overline{T}) &= [\overline{W/Z'}]msig_k \\ &= [\overline{T'/Z}, \overline{W/Z'}]msig_k \\ &\overset{(192)}{=} \langle\overline{X}\rangle\, \overline{U\ x}^n \to U \ \texttt{where}\ \mathcal{P}\end{aligned}$$

as required.

* *Case* $p^? \neq \mathsf{nil}$: Then $p^? = 1$ and $U'' = Z_1$. Hence

$$1 \notin \mathsf{pos}^-(I) \tag{235}$$ {eq:I-not-neg::lemma:mtype-complete}

We now prove that there exists some $\mathcal{R}' \in \mathscr{M}$ such that

$$\Delta \vdash_{\mathrm{a}}{}' \mathsf{impl}(\mathcal{R}', 1) \leq \mathsf{impl}(\mathcal{R}, 1) \text{ for all } \mathcal{R} \in \mathscr{M} \tag{236}$$ {eq:sub-R'2::lemma:mtype-complete}

In the following, we assume w.l.o.g. that $\mathsf{impl}(\mathcal{R}, 1) \neq$ `Object` for all $\mathcal{R} \in \mathscr{M}$. (If $\mathsf{impl}(\mathcal{R}, 1) =$ `Object` then (236) holds trivially for this $\mathcal{R}$.)
Case distinction on whether or not there exists $\mathcal{R}' \in \mathscr{M}$ with $\mathsf{impl}(\mathcal{R}', 1) = L$ for some L.

  · *Case* there exists $\mathcal{R}' \in \mathscr{M}$ with $\mathsf{impl}(\mathcal{R}', 1) = L$ for some L: Then we have $\Delta \Vdash_{\mathrm{q}} L$ `implements` $I\langle\overline{W}\rangle$ by (232). Hence, Lemma 6.32 and (235) give us that $L \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$, so with (233) and Lemma 6.5 we then have

$$\Delta \vdash_{\mathrm{a}}{}' T' \leq I\langle\overline{W}\rangle$$

If $T' = X$ then, by Lemma 11.27, $1 \in \mathsf{pos}^-(I)$, which is a contradiction to (235).
If $T' = N$ then, by Lemma 11.27, $1 \in \mathsf{pos}^-(I)$, which is a contradiction to (235).
Finally, we consider the case where $T' = K'$. Because $\mathsf{impl}(\mathcal{R}, 1) \neq$ `Object` for all $\mathcal{R} \in \mathscr{M}$, we have with (233) and Lemma 6.11 that for all $\mathcal{R} \in \mathscr{M}$:

$$\begin{gathered}\mathsf{impl}(\mathcal{R}, 1) = L_{\mathcal{R}} \text{ for some } L_{\mathcal{R}} \\ K' \trianglelefteq_{\mathrm{i}} L_{\mathcal{R}}\end{gathered} \tag{237}$$ {eq:k'-sub2::lemma:mtype-complete}

With (234), (232), (235), and Lemma 6.32 we get

$$\begin{gathered}L_{\mathcal{R}} \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle \\ 1 \in \mathsf{pos}^+(L_{\mathcal{R}})\end{gathered}$$

With Lemma 6.2 then

$$K' \trianglelefteq_{\mathrm{i}} I\langle\overline{W}\rangle$$

Now assume $1 \in \mathsf{pos}^+(K')$. Then

$$\Delta \Vdash_{\mathrm{q}} K' \ \texttt{implements}\ I\langle\overline{W}\rangle$$

by rule ENT-Q-ALG-ENV and Lemma 6.19. Hence, with (228) and Lemma 11.5

$$K' \ \texttt{implements}\ I\langle\overline{W}\rangle \in \mathscr{M}$$

With (237), we have $\Delta \vdash_{\mathrm{a}}{}' K' \leq L_{\mathcal{R}}$ for all $\mathcal{R} \in \mathscr{M}$, so (236) holds.
On the other hand, assume $1 \notin \mathsf{pos}^+(K')$. Because of (235), we get with Lemma 6.20 that $1 \notin \mathsf{pos}^-(K')$. With (237) and criterion WF-PROG-8 we then have for all $\mathcal{R}_1, \mathcal{R}_2 \in \mathscr{M}$:

$$L_{\mathcal{R}_1} \trianglelefteq_{\mathrm{i}} L_{\mathcal{R}_2} \text{ or } L_{\mathcal{R}_2} \trianglelefteq_{\mathrm{i}} L_{\mathcal{R}_1}$$

With (233) and Lemma 11.12, we know that the set $\{\mathsf{impl}(\mathcal{R}, 1) \mid \mathcal{R} \in \mathscr{M}\}$ is finite. Thus, (236) holds.

· *Case* there does not exist $\mathcal{R}' \in \mathscr{M}$ with $\mathsf{impl}(\mathcal{R}', 1) = L$ for some L: With (233) and Lemma 11.6 we have for all $\mathcal{R}_1, \mathcal{R}_2 \in \mathscr{M}$:

$$L_{\mathcal{R}_1} \trianglelefteq_{\mathrm{i}} L_{\mathcal{R}_2} \text{ or } L_{\mathcal{R}_2} \trianglelefteq_{\mathrm{i}} L_{\mathcal{R}_1}$$

Thus, (236) holds.

End case distinction on whether or not there exists $\mathcal{R}' \in \mathscr{M}$ with $\mathsf{impl}(\mathcal{R}', 1) = L$ for some L.

This finishes the proof of (236). We now use rule PICK-CONSTR-NON-NIL to derive

$$\mathsf{pick\text{-}constr}^{p^?}_{\Delta}\, \mathscr{M} = \mathcal{R}'$$

such that $\Delta \vdash_{\mathrm{a}}{}' \mathsf{impl}(\mathcal{R}', 1) \leq \mathsf{impl}(\mathcal{R}, 1)$ for all $\mathcal{R} \in \mathscr{M}$. We now have by ALG-MTYPE-IFACE (note that $U'' = Z_1$ and, by criterion WF-IFACE-4, $\overline{Z} \cap \mathsf{ftv}(\overline{P''}) = \emptyset$)

$$\begin{aligned}\mathsf{a\text{-}mtype}_{\Delta}(m, T', \overline{T}) &= [\mathsf{impl}(\mathcal{R}', 1)/Z_1, \overline{W/Z'}](\langle \overline{X} \rangle\, \overline{U''\, x} \to U'' \texttt{ where } \overline{P''}) \\ &\overset{(192),(227)}{=} \langle \overline{X} \rangle\, \overline{U\, x} \to \mathsf{impl}(\mathcal{R}', 1) \texttt{ where } \overline{P}\end{aligned}$$

Define $U' = \mathsf{impl}(\mathcal{R}', 1)$. With (229), (230), and (236) we have

$$\Delta \vdash_{\mathrm{a}} \mathsf{impl}(\mathcal{R}', 1) \leq K$$

By (192) and (225) we have

$$U = T_1' = T = K$$

Hence,

$$\Delta \vdash U' \leq U$$

W.l.o.g., $\overline{X} \cap \mathsf{ftv}(\overline{T'}, \mathcal{R}') = \emptyset$. Hence

$$\Delta \vdash \sigma U' \leq \sigma U$$

as required.

End case distinction on the form of $p^?$.

End case distinction on the possibilities left by Lemma 6.32.

End case distinction on the form of m. □

Lemma 11.31. *Assume*

$$\begin{gathered}\texttt{interface } I\langle \overline{Z} \rangle\, [\overline{Y} \texttt{ where } \overline{R}] \texttt{ where } \overline{Q}\, \{\, \overline{m : \texttt{static}\ msig}\ \overline{rcsig}\, \} \\ msig = \langle \overline{X} \rangle\, \overline{U\, x} \to U \texttt{ where } \overline{P} \\ \Delta \Vdash \overline{T} \texttt{ implements } I\langle \overline{W} \rangle \\ \Delta \Vdash [\overline{V/X}][\overline{T/Y}, \overline{W/Z}]\overline{P} \\ \Delta \vdash \overline{T}, \overline{V}\ \mathsf{ok}\end{gathered}$$

such that either $msig \in \overline{msig}$ *or that there exists* $\texttt{receiver}\,\{\overline{m' : msig'}\} \in \overline{rcsig}$ *with* $msig \in \overline{msig'}$*. Then* $\Delta \vdash [\overline{V/X}][\overline{T/Y}, \overline{W/Z}]U$ ok.

Proof. We get with Lemma 11.24 and the assumptions $\Delta \Vdash \overline{T}\ \texttt{implements}\ I\langle\overline{W}\rangle$ and $\Delta \vdash \overline{T}\ \mathsf{ok}$ that

$$\Delta \vdash \overline{T}\ \texttt{implements}\ I\langle\overline{W}\rangle\ \mathsf{ok}$$

Hence,

$$\Delta \vdash \overline{W}\ \mathsf{ok}$$
$$\Delta \Vdash [\overline{T/Y}, \overline{W/Z}]\overline{R}, \overline{Q} \quad (238) \quad \{\texttt{eq:entails-rq::lemma:msig-iface-return-type}\}$$

Because the underlying program is well-typed, we have

$$\overline{R}, \overline{Q}, \overline{Y}, \overline{Z}, \overline{P}, \overline{X} \vdash U\ \mathsf{ok} \quad (239) \quad \{\texttt{eq:u-ok::lemma:msig-iface-return-type}\}$$

W.l.o.g., $\overline{X} \cap \mathsf{ftv}(\overline{R}, \overline{Q}, \overline{T}, \overline{W}) = \emptyset$. Hence,

$$[\overline{T/Y}, \overline{W/Z}]\overline{R}, \overline{Q} = [\overline{V/X}, \overline{T/Y}, \overline{W/Z}]\overline{R}, \overline{Q} \quad (240) \quad \{\texttt{eq:eq1::lemma:msig-iface-return-type}\}$$
$$[\overline{V/X}][\overline{T/Y}, \overline{W/Z}]\overline{P} = [\overline{V/X}, \overline{T/Y}, \overline{W/Z}]\overline{P} \quad (241) \quad \{\texttt{eq:eq2::lemma:msig-iface-return-type}\}$$
$$[\overline{V/X}][\overline{T/Y}, \overline{W/Z}]U = [\overline{V/X}, \overline{T/Y}, \overline{W/Z}]U \quad (242) \quad \{\texttt{eq:eq3::lemma:msig-iface-return-type}\}$$

With (240) and (238) we then have

$$\Delta \Vdash [\overline{V/X}, \overline{T/Y}, \overline{W/Z}]\overline{R}, \overline{Q}$$

With (241) and the assumption $\Delta \Vdash [\overline{V/X}][\overline{T/Y}, \overline{W/Z}]\overline{P}$ we have

$$\Delta \Vdash [\overline{V/X}, \overline{T/Y}, \overline{W/Z}]\overline{P}$$

With Lemma 7.3 and (239) we then have

$$\Delta \vdash [\overline{V/X}, \overline{T/Y}, \overline{W/Z}]U\ \mathsf{ok}$$

so the claim follows with (242). □

Lemma 11.32. *Suppose* $\vdash \Delta\ \mathsf{ok}$ *and and* $\Delta \vdash T_j\ \mathsf{ok}$ *for all* $j \in \mathsf{disp}(I)$. *If* $\Delta \Vdash^?_{\mathrm{a}} \overline{T^?}\ \texttt{implements}\ I\langle\overline{V^?}\rangle \rightharpoonup \overline{T}\ \texttt{implements}\ I\langle\overline{V}\rangle$ *and* $T^?_i = \mathsf{nil}$ *then* $\Delta \vdash T_i\ \mathsf{ok}$.

Proof. We first note that

$$\Delta; \beta; J \vdash^?_{\mathrm{a}} \overline{T^?} \uparrow \overline{U} \rightharpoonup \overline{V} \text{ and } T_j = \mathsf{nil} \text{ imply } V_j = U_j \quad (243) \quad \{\texttt{eq:lift::lemma:entails-questionmark-ok}\}$$
$$\Delta; \beta; J \vdash^?_{\mathrm{a}} \overline{T^?} \uparrow \overline{U} \rightharpoonup \overline{V} \text{ implies } \Delta \vdash_{\mathrm{a}}' V_i \le U_i \text{ for all } i \quad (244) \quad \{\texttt{eq:lift2::lemma:entails-questionmark-ok}\}$$

Now we show that

> *If* $\vdash \Delta\ \mathsf{ok}$ *and* $\Delta \vdash T_j\ \mathsf{ok}$ *for all* $j \in \mathsf{disp}(I)$ *and* $\mathcal{D}{::}\Delta; \mathscr{G}; \beta \Vdash^?_{\mathrm{a}} \overline{T^?}^n\ \texttt{implements}\ I\langle\overline{V^?}\rangle \rightharpoonup \overline{T}\ \texttt{implements}\ I\langle\overline{V}\rangle$ *and* $T^?_i = \mathsf{nil}$ *then* $\Delta \vdash T_i\ \mathsf{ok}$.

Assume $T^?_i = \mathsf{nil}$. W.l.o.g., $i \notin \mathsf{disp}(I)$.
Case distinction on the last rule of $\mathcal{D}$.

- *Case* rule ent-nil-alg-env: Then

  $$R \in \Delta$$
  $$\overline{G}\ \texttt{implements}\ I\langle\overline{V}\rangle \in \mathsf{sup}(R)$$
  $$\Delta; \beta; I \vdash^?_{\mathrm{a}} \overline{T^?} \uparrow \overline{G} \rightharpoonup \overline{T}$$

  From the assumption $\vdash \Delta\ \mathsf{ok}$ and Lemma 11.23 we get $\Delta \vdash \overline{G}\ \texttt{implements}\ I\langle\overline{V}\rangle\ \mathsf{ok}$, so $\Delta \vdash G_i\ \mathsf{ok}$. But with (243) we have $T_i = G_i$.

- *Case* rule ENT-NIL-ALG-IFACE$_1$: Impossible because $n = 1$ and $T_1 \neq \mathsf{nil}$ in this rule.
- *Case* rule ENT-NIL-ALG-IFACE$_2$: Impossible because $n = 1$ and $T_1 \neq \mathsf{nil}$ in this rule.
- *Case* rule ENT-NIL-ALG-IMPL: Then

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V'}\rangle\ [\overline{N}]\ \texttt{where}\ \overline{P}\ \ldots$$

$$\Delta; \beta; I \vdash^{?}_{\mathrm{a}} \overline{T^?} \uparrow [\overline{U/X}]\overline{N} \rightharpoonup \overline{T} \qquad (245)\ \{\texttt{eq:lift3::lemma:entails-questionmark-ok}\}$$

$$\Delta; \mathscr{G} \cup \{[\overline{U/X}]\overline{N}\ \texttt{implements}\ I\langle[\overline{U/X}]\overline{V'}\rangle\}; \texttt{false} \Vdash_{\mathrm{a}} [\overline{U/X}]\overline{P} \qquad (246)\ \{\texttt{eq:entails::lemma:entails-questionmark-ok}\}$$

With (245) and (244) we get

$$(\forall i)\ \Delta \vdash T_i \leq [\overline{U/X}]N_i$$

Thus, if $j \in \mathsf{disp}(I)$ then $\Delta \vdash T_j\ \mathsf{ok}$ by assumption, so with Lemma 11.22

$$\Delta \vdash_{\mathrm{a}}{}'\ [\overline{U/X}]N_j\ \mathsf{ok}$$

From criterion WF-IMPL-1 we get $\overline{X} \subseteq \mathsf{ftv}(\{N_j \mid j \in \mathsf{disp}(I)\})$. Thus, withLemma 7.38,

$$\Delta \vdash \overline{U}\ \mathsf{ok}$$

With Lemma 9.13, (246), and rule ENT-Q-ALG-UP, we get

$$\Delta \Vdash_{\mathrm{q}} [\overline{U/X}]\overline{P}$$

Because the underlying program is well-typed we have

$$\overline{P}, \overline{X} \vdash \overline{N}\ \texttt{implements}\ I\langle\overline{V'}\rangle\ \mathsf{ok}$$

Now Lemma 7.3 yields

$$\Delta \vdash [\overline{U/X}](\overline{N}\ \texttt{implements}\ I\langle\overline{V'}\rangle)\ \mathsf{ok}$$

Thus,

$$\Delta \vdash [\overline{U/X}]\overline{N}\ \mathsf{ok}$$

But with (243) and (245) we have $T_i = [\overline{U/X}]N_i$.

End case distinction on the last rule of $\mathcal{D}$. □

Lemma 11.33. *Suppose* $\vdash \Delta\ \mathsf{ok}$ *and* $\Delta \vdash N, \overline{V}\ \mathsf{ok}$ *and* $\Delta \Vdash [\overline{V/X}]\mathcal{P}$. *If* $\mathsf{a\text{-}mtype}^{\mathrm{c}}(m, N) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}$ *then* $\Delta \vdash [\overline{V/X}]U\ \mathsf{ok}$.

PROOF. By induction on the derivation $\mathcal{D}$ of $\mathsf{a\text{-}mtype}^{\mathrm{c}}(m, N) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}$.
Case distinction on the last rule of $\mathcal{D}$.

- *Case* rule ALG-MTYPE-DIRECT: Then

$$N = C\langle\overline{T}\rangle$$

$$\texttt{class}\ C\langle\overline{Y}\rangle\ \texttt{extends}\ M\ \texttt{where}\ \overline{Q}\,\{\ldots\ \overline{m : msig\,\{e\}}\,\}$$

$$m = m_j$$

$$\langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}} = [\overline{T/Y}]msig_j$$

Assume

$$msig_j = \langle\overline{X}\rangle\,\overline{U'\,x} \to U'\ \texttt{where}\ \overline{P}$$

Because the underlying program is well-typed, we have

$$\overline{Q}, \overline{Y} \vdash m_j : msig_j \, \{e_j\} \; \mathsf{ok}\,\mathsf{in}\; C\langle\overline{Y}\rangle$$

Hence,

$$\overline{Q}, \overline{Y}, \overline{P}, \overline{X} \vdash U' \; \mathsf{ok} \tag{247}$$ {eq:u'-ok::lemma:cache-subset-goalset+cand}

From $\Delta \vdash N \; \mathsf{ok}$ we get $\Delta \Vdash [\overline{T/Y}]\overline{Q}$ and $\Delta \vdash \overline{T} \; \mathsf{ok}$. W.l.o.g., $\overline{X} \cap \mathsf{ftv}(\overline{T}, \overline{Q}) = \emptyset$. Hence, $[\overline{T/Y}]\overline{Q} = [\overline{V/X}, \overline{T/Y}]\overline{Q}$, so we have

$$\Delta \Vdash [\overline{V/X}, \overline{T/Y}]\overline{Q}$$

Moreover, the assumption $\Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}}$ can be written as

$$\Delta \Vdash [\overline{V/X}, \overline{T/Y}]\overline{P}$$

Using Lemma 7.3 on (247) yields

$$\Delta \vdash \underbrace{[\overline{V/X}, \overline{T/Y}]U'}_{=[\overline{V/X}]U} \; \mathsf{ok}$$

as required.

- *Case* rule ALG-MTYPE-SUPER: Then

$$\begin{gathered}
\texttt{class}\; C\langle\overline{X}\rangle \;\texttt{extends}\; M \ldots \\
\mathsf{a\text{-}mtype}^{\mathrm{c}}(m, [\overline{T/X}]M) = \langle\overline{X}\rangle\, \overline{U\,x} \to U \;\texttt{where}\; \overline{\mathcal{P}} \\
N = C\langle\overline{T}\rangle
\end{gathered}$$

  Then $N \trianglelefteq_{\mathrm{c}} [\overline{T/X}]M$, so we get with $\Delta \vdash N \; \mathsf{ok}$ and Lemma 7.16 that $\Delta \vdash [\overline{T/X}]M \; \mathsf{ok}$. The claim now follows from the I.H.

End case distinction on the last rule of $\mathcal{D}$. □

Lemma 11.34. *Suppose* $\vdash \Delta \; \mathsf{ok}$ *and* $\Delta \vdash T, \overline{T}, \overline{V} \; \mathsf{ok}$ *and* $\Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}}$. *If* $\mathsf{a\text{-}mtype}_\Delta(m, T, \overline{T}) = \langle\overline{X}\rangle\, \overline{U\,x} \to U \;\texttt{where}\; \overline{\mathcal{P}}$ *then* $\Delta \vdash [\overline{V/X}]U \; \mathsf{ok}$.

PROOF. *Case distinction* on the rule used to derive $\mathsf{a\text{-}mtype}_\Delta(m, T, \overline{T}) = \ldots$.

- *Case* rule ALG-MTYPE-CLASS: Then $\mathsf{bound}_\Delta(T) = N$ and $\mathsf{a\text{-}mtype}^{\mathrm{c}}(m, N) = \langle\overline{X}\rangle\, \overline{U\,x} \to U \;\texttt{where}\; \overline{\mathcal{P}}$. With Lemma 11.25 we have $\Delta \vdash N \; \mathsf{ok}$. The claim now follows with Lemma 11.33.

- *Case* rule ALG-MTYPE-IFACE: Then

$$\begin{gathered}
\texttt{interface}\; I\langle\overline{Z'}\rangle\, [\overline{Z}^{l} \;\texttt{where}\; \overline{R}] \;\texttt{where}\; \overline{P}\, \{\ldots\; \overline{rcsig}\,\} \\
rcsig_j = \texttt{receiver}\, \{\overline{m : msig}\} \\
msig_k = \langle\overline{X}\rangle\, \overline{U'\,x} \to U' \;\texttt{where}\; \overline{Q} \\
(\forall i \in [l], i \neq j)\; \mathsf{contrib}'_{\Delta;Z_i}(\overline{U}, \overline{T}) = \mathscr{V}_i^{?} \\
\mathsf{contrib}'_{\Delta;Z_j}(Z_j\,\overline{U}, T\,\overline{T}) = \mathscr{V}_j^{?} \\
p^{?} = (\text{if } U = Z_i \text{ for some } i \in [l] \text{ then } i \text{ else } \mathsf{nil}) \\
\overline{W} \;\texttt{implements}\; I\langle\overline{W'}\rangle = \\
\mathsf{pick\text{-}constr}_\Delta^{p^{?}}\{\overline{V''} \;\texttt{implements}\; I\langle\overline{V'''}\rangle \mid (\forall i \in [l]) \text{ if } \mathscr{V}_i^{?} = \mathsf{nil} \text{ then } V_i^{?} = \mathsf{nil} \\
\text{else define } V_i^{?} \text{ such that} \\
\Delta \vdash_{\mathrm{a}}' V_i' \leq V_i^{?} \text{ for } V_i' \in \mathscr{V}_i^{?}, \\
\Delta \Vdash_{\mathrm{a}}^{?} \overline{V^{?}} \;\texttt{implements}\; I\langle\overline{\mathsf{nil}}\rangle \multimap \overline{V''} \;\texttt{implements}\; I\langle\overline{V'''}\rangle\}
\end{gathered}$$

and

$$m = m_k$$
$$\langle \overline{X} \rangle\, \overline{U\,x} \to U \texttt{ where } \overline{\mathcal{P}} = [\overline{W/Z}, \overline{W'/Z'}]msig_k$$

With Lemma 11.22 and the assumption $\Delta \vdash T, \overline{T}$ ok we easily verify that, if $\mathscr{V}_i^? \neq \mathsf{nil}$, then $\Delta \vdash V_i'$ ok for all $V_i' \in \mathscr{V}_i^?$. Hence, we have with Lemma 11.22 for the $V_i^?$ in the argument to $\mathsf{pick\text{-}constr}_\Delta^{p^?}$ that

$$V_i^? \neq \mathsf{nil} \text{ implies } \Delta \vdash V_i^? \;\mathsf{ok}$$

Then, by Lemma 11.26, we have for the V_i'' in the argument to $\mathsf{pick\text{-}constr}_\Delta^{p^?}$

$$V_i^? \neq \mathsf{nil} \text{ implies } \Delta \vdash V_i'' \;\mathsf{ok}$$

Clearly, $\mathscr{V}_i^? \neq \mathsf{nil}$ for all $i \in \mathsf{disp}(I)$, so $V_i^? \neq \mathsf{nil}$ for all $i \in \mathsf{disp}(I)$. Hence, with Lemma 11.32

$$V_i^? = \mathsf{nil} \text{ implies } \Delta \vdash V_i'' \;\mathsf{ok}$$

Hence,

$$\Delta \vdash \overline{W} \;\mathsf{ok}$$

With Lemma 11.3

$$\Delta \Vdash \overline{W} \texttt{ implements } I\langle \overline{W'} \rangle$$

We have $[\overline{V/X}]\overline{\mathcal{P}} = [\overline{V/X}][\overline{W/Z}, \overline{W'/Z'}]\overline{Q}$, so with the assumption $\Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}}$ and Lemma 11.31

$$\Delta \vdash [\overline{V/X}]\, \underbrace{[\overline{W/Z}, \overline{W'/Z'}]U'}_{=U} \;\mathsf{ok}$$

as required.

End case distinction on the rule used to derive $\mathsf{a\text{-}mtype}_\Delta(m, T, \overline{T}) = \ldots$. ☐

Lemma 11.35. *If* $\Delta \vdash N$ ok *and* $\mathsf{fields}(N) = \overline{U\,f}^n$, *then* $\Delta \vdash U_i$ ok *for all* $i \in [n]$.

Proof. We proceed by induction on the derivation of $\mathsf{fields}(N) = \overline{U\,f}^n$.
Case distinction on the last rule in the derivation of $\mathsf{fields}(N) = \overline{U\,f}^n$.

- *Case* rule FIELDS-OBJECT: Then $n = 0$ and the claim holds trivially.
- *Case* rule FIELDS-CLASS: Then $N = C\langle \overline{V} \rangle$ and

$$\texttt{class } C\langle \overline{X} \rangle \texttt{ extends } M \texttt{ where } \overline{P}\, \{\, \overline{T\,f} \ldots \}$$
$$\mathsf{fields}([\overline{V/X}]M) = \overline{T'\,f'}$$
$$\overline{U\,f}^n = \overline{T'\,f'}, [\overline{V/X}]\overline{T\,f}$$

Clearly, $N \trianglelefteq_c [\overline{V/X}]M$, so $\Delta \vdash [\overline{V/X}]M$ ok by Lemma 7.16. Hence, we have by the I.H. that

$$\Delta \vdash \overline{T'} \;\mathsf{ok}$$

The underlying program is well-typed, so we have $\overline{P}, \overline{X} \vdash \overline{T}$ ok. From $\Delta \vdash C\langle \overline{V} \rangle$ ok we get $\Delta \Vdash [\overline{V/X}]\overline{P}$ and $\Delta \vdash \overline{V}$ ok. Hence, with Lemma 7.3,

$$\Delta \vdash [\overline{V/X}]\overline{T} \;\mathsf{ok}$$

End case distinction on the last rule in the derivation of $\mathsf{fields}(N) = \overline{U\,f}^n$. $\square$

Lemma 11.36 (Expression typing ensures well-formedness)**.** *Suppose* $\vdash \Delta$ ok *and* $\Delta \vdash \Gamma$ ok. *If* $\Delta; \Gamma \vdash_{\mathrm{a}} e : T$ *then* $\Delta \vdash T$ ok.

Proof. We proceed by induction on the derivation of $\Delta; \Gamma \vdash_{\mathrm{a}} e : T$.
Case distinction on the last rule used in the derivation of $\Delta; \Gamma \vdash_{\mathrm{a}} e : T$.

- *Case* rule exp-alg-var: Follows with the assumption $\Delta \vdash \Gamma$ ok.
- *Case* rule exp-alg-field: Then

$$\begin{gathered}\Delta; \Gamma \vdash_{\mathrm{a}} e' : T' \\ \mathsf{bound}_\Delta(T') = N \\ \mathsf{fields}(N) = \overline{U\,f} \\ e = e'.f_j \\ T = U_j\end{gathered}$$

  We get from the I.H. that $\Delta \vdash T'$ ok. With Lemma 11.25 then $\Delta \vdash N$ ok. Then we get with Lemma 11.35 that $\Delta \vdash U_j$ ok.

- *Case* rule exp-alg-invoke-d: Then

$$\begin{gathered}e = e'.m\langle\overline{V}\rangle(\overline{e}) \\ T = [\overline{V/X}]U \\ \Delta; \Gamma \vdash_{\mathrm{a}} e' : T' \\ (\forall i)\ \Delta; \Gamma \vdash_{\mathrm{a}} e_i : T_i \\ \mathsf{a\text{-}mtype}_\Delta(m, T', \overline{T}) = \langle\overline{X}\rangle\,\overline{U\,x} \to U \texttt{ where } \overline{\mathcal{P}} \\ \Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}} \\ \Delta \vdash \overline{V}\ \mathsf{ok}\end{gathered}$$

  Applying the I.H. yields $\Delta \vdash T', \overline{T}$ ok, so we can apply Lemma 11.34 and get $\Delta \vdash [\overline{V/X}]U$ ok, as required.

- *Case* rule exp-alg-invoke-s: Then

$$\begin{gathered}e = I\langle\overline{W}\rangle[\overline{T}].m\langle\overline{V}\rangle(\overline{e}) \\ T = [\overline{V/X}]U \\ \Delta \vdash \overline{T}, \overline{V}\ \mathsf{ok} \\ \Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}} \\ \mathsf{a\text{-}smtype}_\Delta(m, I\langle\overline{W}\rangle[\overline{T}]) = \langle\overline{X}\rangle\,\overline{U\,x} \to U \texttt{ where } \overline{\mathcal{P}}\end{gathered}$$

  Applying Lemma 11.34 yields $\Delta \vdash [\overline{V/X}]U$ ok, as required.

- *Case* rule exp-alg-new: Then $\Delta \vdash T$ ok from the premise of this rule.
- *Case* rule exp-alg-cast: Then $\Delta \vdash T$ ok from the premise of this rule.

End case distinction on the last rule used in the derivation of $\Delta; \Gamma \vdash_{\mathrm{a}} e : T$. $\square$

Lemma 11.37. *If* $\mathsf{fields}(N) = \overline{T\,f}^n$ *and* $i \in [n]$, *then there exists* `class` $C\langle\overline{X}\rangle \ldots \{\overline{V\,g} \ldots\}$ *such that* $N \trianglelefteq_{\mathrm{c}} C\langle\overline{U}\rangle$ *and* $T_i\,f_i = [\overline{U/X}]V_j\,g_j$ *for some* j.

PROOF. We proceed by induction on the derivation of $\mathsf{fields}(N) = \overline{T\,f}^n$. The derivation cannot end with rule FIELDS-OBJECT because this would contradict $i \in [n]$. Hence, the last rule must be FIELDS-CLASS. We get

$$\begin{gathered}
N = D\langle\overline{W}\rangle \\
\texttt{class}\ D\langle\overline{X}\rangle\ \texttt{extends}\ M\ \texttt{where}\ \overline{P}\,\{\,\overline{T'\,f'}\ldots\} \\
\mathsf{fields}([\overline{W/X}]M) = \overline{T''\,f''} \\
\overline{T\,f}^n = \overline{T''\,f''}^m, [\overline{W/X}]\overline{T'\,f'}
\end{gathered}$$

If $i > m$ set $C\langle\overline{U}\rangle = D\langle\overline{W}\rangle$. Otherwise, the claim follows with the I.H., the fact that $D\langle\overline{W}\rangle \trianglelefteq_c [\overline{W/X}]M$, and Lemma 6.2. $\square$

Lemma 11.38. *Assume* $\vdash \Delta\ \mathsf{ok}$ *and* $\Delta \vdash T, \overline{T}\ \mathsf{ok}$. *If* $\mathsf{a\text{-}mtype}_\Delta(m, T, \overline{T}) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}$ *then there exists* T' *such that* $\Delta \vdash T \leq T'$ *and* $\mathsf{mtype}_\Delta(m, T') = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}$.

PROOF. *Case distinction* on the form of m.

- *Case* $m = m^{\mathrm{c}}$: Then

$$\begin{gathered}
\mathsf{bound}_\Delta(T) = N \\
\mathcal{D} :: \mathsf{a\text{-}mtype}^{\mathrm{c}}(m, N) = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}
\end{gathered}$$

  A straightforward induction on the derivation $\mathcal{D}$ shows that there exists N' such that

$$\begin{gathered}
\mathsf{mtype}(m, N') = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}} \\
N \trianglelefteq_c N'
\end{gathered}$$

  With $\mathsf{bound}_\Delta(T) = N$ and Lemma 11.17 we have $\Delta \vdash T \leq N$. Thus, by transitivity of subtyping,

$$\Delta \vdash T \leq N'$$

  We finish this case by setting $T' = N'$.

- *Case* $m = m^{\mathrm{i}}$: Then

$$\begin{gathered}
\texttt{interface}\ I\langle\overline{Z'}\rangle\,[\overline{Z}^l\ \texttt{where}\,\overline{R}]\ \texttt{where}\ \overline{P}\,\{\ldots\ \overline{rcsig}\,\} \\
rcsig_j = \texttt{receiver}\,\{\overline{m : msig}\} \\
msig_k = \langle\overline{X}\rangle\,\overline{U'\,x} \to U'\ \texttt{where}\ \overline{Q} \\
(\forall i \in [l], i \neq j)\ \mathsf{contrib}'_{\Delta;Z_i}(\overline{U'}, \overline{T}) = \mathscr{V}_i^? \\
\mathsf{contrib}'_{\Delta;Z_j}(Z_j\,\overline{U'}, T\,\overline{T}) = \mathscr{V}_j^? \\
p^? = (\text{if } U' = Z_i \text{ for some } i \in [l] \text{ then } i \text{ else } \mathsf{nil}) \\
\overline{W}\ \texttt{implements}\ I\langle\overline{W'}\rangle = \mathsf{pick\text{-}constr}_\Delta^{p^?}\,\mathscr{M} \\
\begin{aligned}
\mathscr{M} = \{\overline{V}\ \texttt{implements}\,I\langle\overline{V''}\rangle \mid (\forall i \in [l])\ &\text{if } \mathscr{V}_i^? = \mathsf{nil} \text{ then } V_i^? = \mathsf{nil} \\
&\text{else define } V_i^? \text{ such that} \\
&\quad \Delta \vdash_{\mathrm{a}}{}' V_i' \leq V_i^? \text{ for } V_i' \in \mathscr{V}_i^?, \\
&\Delta \Vdash_{\mathrm{a}}^? \overline{V^?}\ \texttt{implements}\,I\langle\overline{\mathsf{nil}}\rangle \twoheadrightarrow \overline{V}\ \texttt{implements}\,I\langle\overline{V''}\rangle\}
\end{aligned}
\end{gathered}$$

  and

$$\langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}} = [\overline{W/Z}, \overline{W'/Z'}](\langle\overline{X}\rangle\,\overline{U'\,x} \to U'\ \texttt{where}\ \overline{Q})$$

Obviously, $\mathscr{V}^?_j \neq \mathsf{nil}$. With Lemma 11.13 and the definition of $\mathsf{contrib}'$ we get

$$\Delta \vdash_{\mathrm{a}}{}' T \leq V'_j \text{ for all } V'_j \in \mathscr{V}^?_j$$

With Lemma 11.26, we know that for all $\overline{V}\,\texttt{implements}\,I\langle\overline{V''}\rangle \in \mathscr{M}$ there exists some $V'_j \in \mathscr{V}^?_j$ such that

$$\Delta \vdash_{\mathrm{a}}{}' V'_j \leq V_j$$

With rule SUB-TRANS we thus have

$$\Delta \vdash T \leq W_j$$

By Lemma 11.3 we get

$$\Delta \Vdash \overline{W}\,\texttt{implements}\,I\langle\overline{W'}\rangle$$

By rule MTYPE-IFACE we now have

$$\mathsf{mtype}_\Delta(m, W_j) = [\overline{W/Z}, \overline{W'/Z'}]msig_k = \langle\overline{X}\rangle\,\overline{U\,x} \to U\ \texttt{where}\ \overline{\mathcal{P}}$$

Define $T' = W_j$ to finish this case.

End case distinction on the form of m. □

Theorem 11.39 (Soundness of algorithmic expression typing)**.** *Suppose* $\vdash \Delta\ \mathsf{ok}$ *and* $\Delta \vdash \Gamma\ \mathsf{ok}$. *If* $\Delta; \Gamma \vdash_{\mathrm{a}} e : T$ *then* $\Delta; \Gamma \vdash e : T$.

PROOF. We proceed by induction on the derivation of $\Delta; \Gamma \vdash_{\mathrm{a}} e : T$.
Case distinction on the last rule of the derivation of $\Delta; \Gamma \vdash_{\mathrm{a}} e : T$.

- *Case* rule EXP-ALG-VAR: Obvious.
- *Case* rule EXP-ALG-FIELD: Inverting the rule yields

$$\begin{gathered} e = e'.f_j \\ \Delta; \Gamma \vdash_{\mathrm{a}} e' : T' \\ \mathsf{bound}_\Delta(T') = N \\ \mathsf{fields}(N) = \overline{U\,f} \\ T = U_j \end{gathered}$$

With Lemma 11.37 there exists a class C such that

$$\begin{gathered} \texttt{class}\ C\langle\overline{X}\rangle \ldots \{\overline{V\,g} \ldots\} \\ N \trianglelefteq_{\mathrm{c}} C\langle\overline{W}\rangle \\ U_j\,f_j = [\overline{W/X}]V_i\,g_i \end{gathered} \tag{248}$$

{eq:1::lemma:soundness-alg-exp-typing}

By Lemma 11.17 we have $\Delta \vdash T' \leq N$, so $\Delta \vdash T' \leq C\langle\overline{W}\rangle$. We get by the I.H. that $\Delta; \Gamma \vdash e' : T'$, so with rule EXP-SUBSUME, $\Delta; \Gamma \vdash e' : C\langle\overline{W}\rangle$. The claim now follows with rule EXP-FIELD and (248).

- *Case* rule EXP-ALG-INVOKE-D: We get from the premises of the rule

$$e = e'.m\langle\overline{V}\rangle(\overline{e})$$
$$T = [\overline{V/X}]U$$
$$\Delta;\Gamma \vdash_{\mathrm{a}} e' : T'$$
$$(\forall i)\ \Delta;\Gamma \vdash_{\mathrm{a}} e_i : T_i$$
$$\textsf{a-mtype}_\Delta(m, T', \overline{T}) = \langle\overline{X}\rangle\,\overline{U\,x} \to U \ \texttt{where}\ \overline{\mathcal{P}}$$
$$(\forall i)\ \Delta \vdash_{\mathrm{a}} T_i \leq [\overline{V/X}]U_i$$
$$\Delta \Vdash_{\mathrm{a}} [\overline{V/X}]\overline{\mathcal{P}}$$
$$\Delta \vdash_{\mathrm{a}} \overline{V}\ \textsf{ok}$$

  By the I.H.

$$\Delta;\Gamma \vdash e' : T'$$
$$(\forall i)\ \Delta;\Gamma \vdash e_i : T_i$$

  With Lemma 11.36

$$\Delta \vdash T', \overline{T}\ \textsf{ok}$$

  With Lemma 11.38, we get the existence of T'' such that

$$\Delta \vdash T' \leq T''$$
$$\textsf{mtype}_\Delta(m, T'') = \langle\overline{X}\rangle\,\overline{U\,x} \to U \ \texttt{where}\ \overline{\mathcal{P}}$$

  We have by rule EXP-SUBSUME

$$\Delta;\Gamma \vdash e' : T''$$
$$(\forall i)\ \Delta;\Gamma \vdash e_i : [\overline{V/X}]U_i$$

  so the claim follows with rule EXP-INVOKE.

- *Case* rule EXP-ALG-INVOKE-S: We use the I.H. and rule EXP-SUBSUME to derive the correct types for the arguments of the call. With Corollary 11.2, we get that smtype and a-smtype are equivalent. The claim then follows with rule EXP-INVOKE-S.

- *Case* rule EXP-ALG-NEW: We use the I.H. and rule EXP-SUBSUME to derive the correct types for the arguments of the constructor call. The claim then follows with rule EXP-NEW.

- *Case* rule EXP-ALG-CAST: Follows from the I.H.

End case distinction on the last rule of the derivation of $\Delta;\Gamma \vdash_{\mathrm{a}} e : T$. □

Lemma 11.40. *If* `class` $C\langle\overline{X}\rangle \ldots \{\overline{U\,f} \ldots\}$ *and* $N \trianglelefteq_{\mathrm{c}} C\langle\overline{T}\rangle$ *then* $\textsf{fields}(N) = \ldots \overline{U'\,f} \ldots$ *such that* $[\overline{T/X}]\overline{U} = \overline{U'}$.

PROOF. Follows by a routine induction on the derivation of $N \trianglelefteq_{\mathrm{c}} C\langle\overline{T}\rangle$. □

Theorem 11.41 (Completeness of algorithmic expression typing)**.** *Suppose* $\vdash \Delta$ ok *and* $\Delta \vdash \Gamma$ ok. *If* $\Delta;\Gamma \vdash e : T$ *then* $\Delta;\Gamma \vdash_{\mathrm{a}} e : U$ *such that* $\Delta \vdash U \leq T$.

PROOF. We proceed by induction on the derivation of $\Delta;\Gamma \vdash e : T$.
Case distinction on the last rule used in the derivation of $\Delta;\Gamma \vdash e : T$.

- *Case* rule EXP-VAR: Obvious.

- *Case* rule EXP-FIELD: By inverting the rule, we get

$$\Delta; \Gamma \vdash e' : C\langle\overline{T}\rangle$$
$$\texttt{class } C\langle\overline{X}\rangle \texttt{ extends } N \texttt{ where } \overline{P} \, \{ \, \overline{U\,f} \ldots \}$$
$$e = e'.f_j$$
$$T = [\overline{T/X}]U_j$$

We get from the I.H.

$$\Delta; \Gamma \vdash_{\mathrm{a}} e' : T'$$
$$\Delta \vdash T' \leq C\langle\overline{T}\rangle$$

Hence, with Corollary 11.2,

$$\Delta \vdash_{\mathrm{a}}{}' \, T' \leq C\langle\overline{T}\rangle$$

By Lemma 11.18

$$\mathsf{bound}_\Delta(T') = N$$
$$N \trianglelefteq_{c} C\langle\overline{T}\rangle$$

By Lemma 11.40

$$\mathsf{fields}(N) = \ldots \, \overline{U'\,f} \, \ldots$$
$$[\overline{T/X}]\overline{U} = \overline{U'}$$

The claim now follows with rule EXP-ALG-FIELD.

- *Case* rule EXP-INVOKE: Inverting the rule yields

$$e = e'.m\langle\overline{V}\rangle(\overline{e})$$
$$T = [\overline{V/X}]U'$$
$$\Delta; \Gamma \vdash e' : T'$$
$$(\forall i) \;\; \Delta; \Gamma \vdash e_i : [\overline{V/X}]U_i$$
$$\mathsf{mtype}_\Delta(m, T') = \langle\overline{X}\rangle \, \overline{U\,x} \to U' \texttt{ where } \overline{\mathcal{P}}$$
$$\Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}}$$
$$\Delta \vdash \overline{V} \ \mathsf{ok}$$

By the I.H.

$$\Delta; \Gamma \vdash_{\mathrm{a}} e' : T''$$
$$\Delta \vdash T'' \leq T'$$
$$(\forall i) \;\; \Delta; \Gamma \vdash_{\mathrm{a}} e_i : W_i$$
$$(\forall i) \;\; \Delta \vdash W_i \leq [\overline{V/X}]U_i$$

Now with Lemma 11.36

$$\Delta \vdash T'' \ \mathsf{ok}$$

By Lemma 11.30

$$\mathsf{a\text{-}mtype}_\Delta(m, T'', \overline{W}) = \langle\overline{X}\rangle \, \overline{U'\,x} \to U'' \texttt{ where } \overline{\mathcal{P}}$$
$$(\forall i) \;\; \Delta \vdash W_i \leq [\overline{V/X}]U'_i$$
$$\Delta \vdash [\overline{V/X}]U'' \leq [\overline{V/X}]U'$$

We now get with rule EXP-ALG-INVOKE-D

$$\Delta; \Gamma \vdash_{\mathrm{a}} e'.m\langle\overline{V}\rangle(\overline{e}) : [\overline{V/X}]U''$$

- *Case* rule EXP-INVOKE-S: Inverting the rule yields

$$\begin{gathered} e = I\langle\overline{W}\rangle[\overline{T}].m\langle\overline{V}\rangle(\overline{e}) \\ T = [\overline{V/X}]U' \\ \mathsf{smtype}_\Delta(m, I\langle\overline{W}\rangle[\overline{T}]) = \langle\overline{X}\rangle\, \overline{U\, x} \to U' \ \texttt{where}\ \overline{\mathcal{P}} \\ (\forall i)\ \Delta; \Gamma \vdash e_i : [\overline{V/X}]U_i \\ \Delta \Vdash [\overline{V/X}]\overline{\mathcal{P}} \\ \text{not } 1 \in \mathsf{pos}^+(I) \text{ or } (\exists i)\ \Delta \Vdash T_i\, \texttt{mono} \\ \Delta \vdash \overline{T}, \overline{V}\ \mathsf{ok} \end{gathered}$$

By the I.H.

$$\begin{gathered} (\forall i)\ \Delta; \Gamma \vdash_{\mathrm{a}} e_i : W_i \\ \Delta \vdash W_i \leq [\overline{V/X}]U_i \end{gathered}$$

With Corollary 11.2, we get that smtype and a-smtype are equivalent. We then have by rule EXP-ALG-INVOKE-S

$$\Delta; \Gamma \vdash_{\mathrm{a}} I\langle\overline{W}\rangle[\overline{T}].m\langle\overline{V}\rangle(\overline{e}) : [\overline{V/X}]U'$$

- *Case* rule EXP-NEW: The claim follows from the I.H. and rule EXP-NEW.
- *Case* rule EXP-CAST: The claim follows from the I.H. and rule EXP-CAST.
- *Case* rule EXP-SUBSUME: From the premise of the rule, we get $\Delta; \Gamma \vdash e : U'$ and $\Delta \vdash U' \leq T$. The I.H. yields $\Delta; \Gamma \vdash_{\mathrm{a}} e : U$ and $\Delta \vdash U \leq U'$. We then have $\Delta \vdash U \leq T$ by rule SUB-TRANS.

End case distinction on the last rule used in the derivation of $\Delta; \Gamma \vdash e : T$. □

12 Termination of Expression Typing

Lemma 12.1. *If $T_i^? \neq \mathsf{nil}$ for all $i \in \mathsf{disp}(I)$, then the set $\mathscr{R} = \{\mathcal{R} \mid \Delta \Vdash_{\mathrm{a}}^? \overline{T^?}\ \texttt{implements}\ I\langle\overline{V^?}\rangle \multimap \mathcal{R}\}$ is finite.*

PROOF. We generalize the claim and prove that $\mathscr{R} = \{\mathcal{R} \mid \Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}}^? \overline{T^?}\ \texttt{implements}\ I\langle\overline{V^?}\rangle \multimap \mathcal{R}\}$ is finite. Assume $\mathscr{R} = \{\mathcal{R}_1, \mathcal{R}_2, \dots\}$ is infinite. W.l.o.g., assume for all $i \in \mathbb{N}$

$$\begin{gathered} \mathcal{D}_i :: \Delta; \mathscr{G}; \beta \Vdash_{\mathrm{a}}^? \overline{T^?}\ \texttt{implements}\ I\langle\overline{V^?}\rangle \multimap \mathcal{R}_i \\ i \neq j \text{ implies } \mathcal{R}_i \neq \mathcal{R}_j \end{gathered}$$

such that all $\mathcal{D}_i$ end with the same rule.
Case distinction on the last rule in all $\mathcal{D}_i$.

- *Case* rule ENT-NIL-ALG-ENV: Impossible because Δ is finite and, obviously, $\mathsf{sup}(\mathcal{S})$ is finite for all $\mathcal{S}$.
- *Case* rule ENT-NIL-ALG-IFACE$_1$: Impossible because the set $\{I\langle\overline{V}\rangle \mid \Delta; \beta; I \vdash_{\mathrm{a}} T_1 \uparrow I\langle\overline{V}\rangle\}$ is finite by Lemma 11.12.

- *Case* rule ENT-NIL-ALG-IFACE$_2$: Impossible because the set $\{J\langle\overline{V}\rangle \mid J'\langle\overline{W}\rangle \trianglelefteq_{\mathrm{i}} J\langle\overline{V}\rangle\}$ is finite by Lemma 11.12.
- *Case* rule ENT-NIL-ALG-IMPL: W.l.o.g., assume that the same implementation definition

  $$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{V}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots$$

  appears in the premise of the last rule of every $\mathcal{D}_i$. (There are only finitely many implementation definitions in a program, so infinitely many derivations must share the same implementation definition.) We then have

  $$\mathcal{R}_i = \overline{T}\ \texttt{implements}\ I\langle[\overline{U_i/X}]\overline{V}\rangle$$
  $$\Delta;\beta;I \vdash^?_{\mathrm{a}} \overline{T^?} \uparrow [\overline{U_i/X}]\overline{N} \twoheadrightarrow \overline{T}$$

  Clearly, for $j \in \mathsf{disp}(I)$, we have

  $$\Delta \vdash_{\mathrm{a}}{}'\ T^?_j \leq [\overline{U_i/X}]N_j$$

  With criterion WF-IMPL-1 we have $\overline{X} \subseteq \mathsf{ftv}(\{N_i \mid i \in \mathsf{disp}(I)\})$, so with Lemma 11.12 we know that the set $\{U_i \mid i \in \mathbb{N}\}$ is finite. Hence, the set

  $$\{[\overline{U_i/X}]\overline{N} \mid i \in \mathbb{N}\} \cup \{[\overline{U_i/X}]\overline{V} \mid i \in \mathbb{N}\}$$

  is finite. But if $T^?_j = \mathsf{nil}$ then $T_j = [\overline{U_i/X}]N_j$. Hence, the set $\mathscr{R}$ cannot be infinite, which contradicts our assumption.

End case distinction on the last rule in all $\mathcal{D}_i$. □

Lemma 12.2. *Let*

$$\begin{aligned}\mathscr{M} = \{\overline{V}\ \texttt{implements}\ I\langle\overline{V''}\rangle \mid (\forall i \in [l])\ &\textit{if } \mathscr{V}^?_i = \mathsf{nil} \textit{ then } V^?_i = \mathsf{nil}\\ &\textit{else define } V^?_i \textit{ such that}\\ &\quad \Delta \vdash_{\mathrm{a}}{}'\ V'_i \leq V^?_i \textit{ for } V'_i \in \mathscr{V}^?_i,\\ \Delta \Vdash^?_{\mathrm{a}} \overline{V^?}\ \texttt{implements}\ I\langle\overline{\mathsf{nil}}\rangle &\twoheadrightarrow \overline{V}\ \texttt{implements}\ I\langle\overline{V''}\rangle\}\end{aligned}$$

If $\mathscr{V}^?_i \neq \mathsf{nil}$ for all $i \in \mathsf{disp}(I)$ and all $\mathscr{V}^?_i$ are finite, then $\mathscr{M}$ is finite.

PROOF. With Lemma 11.12 we know that only finitely many choices for the $V^?_i$s in the definition of $\mathscr{M}$ exist. Moreover, $V^?_i \neq \mathsf{nil}$ for all $i \in \mathsf{disp}(I)$. The claim now follows with Lemma 12.1. □

Theorem 12.3. *The problem of finding some type T such that $\Delta;\Gamma \vdash e : T$ is derivable for given Δ, Γ, and e is decidable.*

PROOF. By Theorem 11.39 and Theorem 11.41, it suffices to define a total function $\texttt{type}(\Delta,\Gamma,e)$ that returns type T if, and only if, $\Delta;\Gamma \vdash_{\mathrm{a}} e : T$ is derivable. The definition of `type` is straightforward, so we only need to verify that `type` is a total function; that is, that `type` terminates for all inputs. Clearly, the third argument of a recursive call of `type` is always a subexpression of the original expression argument; hence, there are only finitely many recursive calls of `type`. Similarly, the function checking the relations $\Delta \vdash_{\mathrm{a}} T\ \mathsf{ok}$ and $\Delta \vdash_{\mathrm{a}} \mathcal{P}\ \mathsf{ok}$ calls itself only on strictly smaller arguments. Moreover, the functions `entails` and `sub` for checking entailment and subtyping, respectively, terminate by Theorem 10.22.

The only possible sources of non-termination left are the auxiliaries a-mtype, a-smtype, bound, and fields. Thereof, a-smtype and fields obviously terminate. A call $\mathsf{bound}_\Delta(N)$ or $\mathsf{bound}_\Delta(N)$ clearly terminates. For $\mathsf{bound}_\Delta(X)$, we get by Lemma 11.12 that the set $\{\Delta \vdash_{\mathrm{a}}{}'\ X \leq N\}$ is finite. Thus, a call $\mathsf{bound}_\Delta(X)$ also terminates.

We now consider a call $\mathsf{a\text{-}mtype}(m,T,\overline{T})$. If $m = m^{\mathrm{c}}$, then the call obviously terminates. Otherwise, we check that all premises of rule ALG-MTYPE-IFACE terminate. With Lemma 11.12 we easily verify that all $\mathscr{V}^?_i$ in the premise are finite and that $\mathscr{V}^?_i \neq \mathsf{nil}$ for all $i \in \mathsf{disp}(I)$. By Lemma 12.2 we then have that the argument of pick-constr is finite, so the premise involving pick-constr terminates. The remaining premises terminate trivially. □

```
unify⊓(Δ, X̄, {G11 ⊓? G12,  ...,  Gn1 ⊓? Gn2}) {
    for ( ((i1, j1), ..., (in, jn)) ∈ ∏_{i=1}^{n} {(1,2), (2,1)} ) {
        𝕌 = (Δ, X̄, {G1i1 ≤? G1j1, ..., Gnin ≤? Gnjn});
        if (unify≤(𝕌) == OK(σ))
            return OK(σ);
    }
    return FAIL;
}
```

Figure 21: Algorithm for unification modulo greatest lower bounds

13 Checking the Well-formedness Criteria

It is not obvious how to check well-formedness criteria Wf-Prog-2, Wf-Prog-3, Wf-Prog-4, Wf-Tenv-3, and Wf-Tenv-7(2). In Sec. 13.1, we now show how to check all of these criteria except criterion Wf-Tenv-3. Sec. 13.2 then deals with criterion Wf-Tenv-3.

13.1 Unification and Subtyping

Directly checking well-formedness criteria Wf-Prog-2, Wf-Prog-3, Wf-Prog-4, and Wf-Tenv-7(2) is not possible because these criteria involve universal quantification over one or two substitutions subject to subtype or greatest lower bounds conditions.

Definition 13.1 (Unification modulo greatest lower bounds)**.** *A* unification problem modulo greatest lower bounds *is a triple*

$$\mathbb{U} = (\Delta, \overline{X}, \{G_1 \leq^? H_1, \ldots G_n \leq^? H_n\}) \quad .$$

A solution *of* $\mathbb{U}$ *is a substitution* σ *with* $\mathsf{dom}(\sigma) \subseteq \overline{X}$ *such that* $\Delta \vdash \sigma G_i \sqcap \sigma H_i$ *for all* $i \in [n]$*. We write* $\mathsf{sol}(\mathbb{U})$ *for the* set of all solutions *of* $\mathbb{U}$*. A* most general solution *of* $\mathbb{U}$ *is a solution* σ *such that for any other solution* σ' *it holds that* $\sigma \preceq \sigma'$*. We say that* $\mathbb{U}$ *is* well-formed *iff* $\mathsf{ftv}(\Delta) \cap \overline{X} = \emptyset$ *and* $G_i = Y$ *(or* $H_i = Y$*) implies* $Y \notin \overline{X}$ *for any* $i \in [n]$*.*

Fig. 21 defines an algorithm for solving unification problems modulo greatest lower bounds. By looking at Definition 5.3, we see that a solution of $(\Delta, \overline{X}, \{G_{11} \sqcap^? G_{12}, \ldots, G_{n1} \sqcap^? G_{n2}\})$ must also solve the unification problem modulo subtyping $(\Delta, \overline{X}, \{G_{1i_1} \leq^? G_{1j_1}, \ldots, G_{ni_n} \leq^? G_{nj_n}\})$ for a set of pairs $\{(i_1, j_1), \ldots, (i_n, j_n)\}$ where $(i_k, j_k) \in \{(1,2), (2,1)\}$ for all $k \in [n]$. The pseudo-code in Fig. 21 essentially tries all possible set of pairs.

Lemma 13.2. *Suppose*

$$\mathbb{U} = (\Delta, \overline{X}, \{G_{11} \sqcap^? G_{12}, \ldots, G_{n1} \sqcap^? G_{n2}\})$$
$$((i_1, j_1), \ldots, (i_n, j_n)) \in \prod_{i=1}^{n} \{(1,2), (2,1)\}$$
$$\mathbb{U}' = (\Delta, \overline{X}, \{G_{1i_1} \leq^? G_{1j_1}, \ldots, G_{ni_n} \leq^? G_{nj_n}\})$$

and assume that $\mathbb{U}$ *is well-formed. Then either* $\mathsf{sol}(\mathbb{U}') = \emptyset$ *or* $\mathsf{sol}(\mathbb{U}) = \mathsf{sol}(\mathbb{U}')$*.*

Proof. If $\mathsf{sol}(\mathbb{U}') = \emptyset$, then nothing is to prove. Thus, assume $\mathsf{sol}(\mathbb{U}') \neq \emptyset$.

- "$\mathsf{sol}(\mathbb{U}) \subseteq \mathsf{sol}(\mathbb{U}')$". Assume $\sigma \in \mathsf{sol}(\mathbb{U})$. Then, by Definition 5.3,

$$((i'_1, j'_1), \ldots, (i'_n, j'_n)) \in \prod_{i=1}^{n} \{(1,2), (2,1)\}$$

such that for all $k \in [n]$

$$\Delta \vdash \sigma G_{k i'_k} \leq \sigma G_{k j'_k} \tag{249}$$ {eq:st'::lemma:solution-glb}

From $\mathsf{sol}(\mathbb{U}') \neq \emptyset$ we get the existence of a substitution τ such that for all $k \in [n]$

$$\Delta \vdash \tau G_{k i_k} \leq \sigma G_{k j_k}$$

It is easy to see that, because $\mathbb{U}$ is well-formed, $\mathbb{U}'$ is well-formed. Hence,

$$\mathsf{dom}(\sigma) \subseteq \overline{X} \tag{250}$$ {eq:dom1::lemma:solution-glb}

$$\mathsf{dom}(\tau) \subseteq \overline{X} \tag{251}$$ {eq:dom2::lemma:solution-glb}

We now show $\Delta \vdash \sigma G_{k i_k} \leq \sigma G_{k j_k}$ for all $k \in [n]$. This implies $\sigma \in \mathsf{sol}(\mathbb{U}')$.
Assume $k \in [n]$. We have $(i_k, j_k) = (1, 2)$ or $(i_k, j_k) = (2, 1)$, and $(i'_k, j'_k) = (1, 2)$ or $(i'_k, j'_k) = (2, 1)$. If $(i_k, j_k) = (i'_k, j'_k)$ then with (249) $\Delta \vdash \sigma G_{k i_k} \leq \sigma G_{k j_k}$. Thus, assume $(i_k, j_k) \neq (i'_k, j'_k)$. W.l.o.g., $(i_k, j_k) = (1, 2)$ and $(i'_k, j'_k) = (2, 1)$. Hence, $\Delta \vdash \sigma G_{k2} \leq \sigma G_{k1}$ and $\Delta \vdash \tau G_{k1} \leq \tau G_{k2}$. With (250), (251), and the well-formedness of $\mathbb{U}$, we know that σG_{k2}, σG_{k1}, τG_{k2}, and τG_{k1} are all G-types. Thus, with Theorem 6.34 and Lemma 6.16:

$$\Delta \vdash_{\mathrm{q}}' \sigma G_{k2} \leq \sigma G_{k1}$$
$$\Delta \vdash_{\mathrm{q}}' \tau G_{k1} \leq \tau G_{k2}$$

Case distinction on the form of G_{k2}.

- *Case* $G_{k2} = Y$ for some Y: $\mathbb{U}$ is well-formed, so $Y \notin \overline{X}$. Hence, with (250) and (251), $\sigma G_{k2} = Y = \tau G_{k2}$. With Lemma 6.11 then $\tau G_{k1} = Y$, so $G_{k1} = Y$. Thus, $\Delta \vdash \sigma G_{k i_k} \leq \sigma G_{k j_k}$.
- *Case* $G_{k2} = C\langle\overline{T}\rangle$ for some $C\langle\overline{T}\rangle$: With Lemma 6.11 then $\sigma G_{k1} = \sigma D\langle\overline{U}\rangle$. By inverting rule SUB-Q-ALG-CLASS, we get

  $$\sigma C\langle\overline{T}\rangle \trianglelefteq_{c} \sigma D\langle\overline{U}\rangle$$
  $$\tau D\langle\overline{U}\rangle \trianglelefteq_{c} \tau C\langle\overline{T}\rangle$$

  The class graph is acyclic (criterion WF-PROG-5), so

  $$C = D$$
  $$\sigma\overline{T} = \sigma\overline{U}$$

  Thus, $\Delta \vdash \sigma G_{k1} \leq \sigma G_{k2}$, so $\Delta \vdash \sigma G_{k i_k} \leq \sigma G_{k j_k}$.

End case distinction on the form of G_{k2}.

- "$\mathsf{sol}(\mathbb{U}') \subseteq \mathsf{sol}(\mathbb{U})$". If $\sigma \in \mathsf{sol}(\mathbb{U}')$ then obviously also $\sigma \in \mathsf{sol}(\mathbb{U})$. □

Theorem 13.3 (Soundness, completeness, and termination of $\mathtt{unify}_{\sqcap}(\mathbb{U})$)**.** *Let $\mathbb{U}$ be a well-formed unification problem modulo greatest lower bounds. If $\mathbb{U}$ has a solution then $\mathtt{unify}_{\sqcap}(\mathbb{U})$ returns* ***OK(σ)****, where σ is an idempotent, most general solution of $\mathbb{U}$. If $\mathbb{U}$ does not have a solution, $\mathtt{unify}_{\sqcap}(\mathbb{U})$ returns* ***FAIL****.*

PROOF. Termination of $\mathtt{unify}_{\sqcap}(\mathbb{U})$ follows from Theorem 10.5.
Next, assume $\mathbb{U}$ does not have a solution. Thus, none of the unification problems constructed in line 3 of Fig. 21 has a solution. The claim now follows from Theorem 10.8.
Finally, assume that $\mathbb{U}$ has a solution. Thus, some of the unification problems constructed in line 3 have solutions. Assume that $\mathbb{U}'$ is the first of these problems. According to Lemma 13.2, we then have $\mathsf{sol}(\mathbb{U}) = \mathsf{sol}(\mathbb{U}')$. The claim now follows with Theorem 10.9. □

Lemma 13.4. *If a well-formed unification problem modulo subtyping (or modulo greatest lower bounds) has a solution, than it also has a most general solution.*

PROOF. Follows from Theorems 10.5, 10.8, 10.9, and 13.3. □

Now we are in the position to present alternative formulations of well-formedness criteria WF-PROG-2, WF-PROG-3, WF-PROG-4, and WF-TENV-7(2).

WF-PROG-2′ For each pair of disjoint implementation definitions

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\,\overline{M}\,]\ \texttt{where}\ \overline{P}\ \ldots \qquad \texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{Q}\ \ldots$$

with $\overline{X}\cap\overline{Y}=\emptyset$ and where σ is a most general solution to the unification problem $(\emptyset, \overline{X}\,\overline{Y}, \{M_i \sqcap^? N_i \mid i \in \mathsf{disp}(I)\})$, it holds that $\sigma\overline{T} = \sigma\overline{U}$ and that $\sigma M_j = \sigma N_j$ for all $j \notin \mathsf{disp}(I)$.

WF-PROG-3′ For each pair of disjoint implementation definitions

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\,\overline{N}^n\,]\ \texttt{where}\ \overline{P}\ \ldots$$
$$\texttt{implementation}\langle\overline{X'}\rangle\ I\langle\overline{T'}\rangle\ [\,\overline{N'}^n\,]\ \texttt{where}\ \overline{P'}\ \ldots$$

with $\overline{X}\cap\overline{X'}=\emptyset$ and where σ is a most general solution to the unification problem $(\emptyset, \overline{X}\,\overline{X'}, \{N_i \sqcap^? N'_i \mid i \in [n]\})$, there exists an implementation definition

$$\texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{M}\,]\ \texttt{where}\ \overline{Q}\ \ldots$$

and a substitution $[\overline{W/Y}]$ such that $\emptyset \vdash \sigma\overline{N} \sqcap \sigma\overline{N'} = [\overline{W/Y}]\overline{M}$.

WF-PROG-4′ For each pair of disjoint implementation definitions

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\,\overline{M}\,]\ \texttt{where}\ \overline{P}\ \ldots \qquad \texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{Q}\ \ldots$$

with $\overline{X}\cap\overline{X'}=\emptyset$ and where σ is a most general solution to the unification problem $(\emptyset, \overline{X}\,\overline{Y}, \{M_i \leq^? N_i \mid i \in [n]\})$, it holds that for all $\mathcal{P} \in \sigma\overline{P}$ either $\{Q \in \sigma\overline{Q}\} \Vdash \mathcal{P}$ or $\mathcal{P} \in \sigma\overline{Q} \cup \mathsf{sup}(\sigma\overline{Q}) \cup \{T\ \texttt{extends}\ U \mid T\ \texttt{extends}\ U' \in \sigma\overline{Q}, \{Q \in \sigma\overline{Q}\} \vdash_{\mathrm{q}}{}'\ U' \leq U\}$.

WF-TENV-7′

1. Unchanged from criterion WF-TENV-7.
2. For each constraint and each implementation definition

$$\overline{G}\ \texttt{implements}\ I\langle\overline{T}\rangle \in \mathsf{sup}(\Delta)$$
$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{W}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{P}\ \ldots$$

with $\overline{X}\cap\mathsf{ftv}(\mathsf{sup}(\Delta))=\emptyset$ and where σ is a most general solution to the unification problem $(\Delta, \overline{X}, \{G_i \sqcap^? N_i \mid i \in \mathsf{disp}(I)\})$, it holds that $\overline{T} = \sigma\overline{W}$ and $G_j = \sigma N_j$ for all $j \notin \mathsf{disp}(I) \cup \mathsf{pos}^-(I)$.

Given the algorithms in Fig. 15 and Fig. 21, we can effectively check the criteria just defined.

Lemma 13.5. *Criterion* WF-PROG-2 *and criterion* WF-PROG-2′ *are equivalent.*

PROOF. The proof is easy, using Lemma 13.4 for the implication "⇐". □

Lemma 13.6. *Criterion* WF-PROG-3 *and criterion* WF-PROG-3′ *are equivalent.*

PROOF. The proof is easy, using Lemma 13.4 for the implication "⇐". □

Lemma 13.7. *Criterion* WF-PROG-4′ *implies criterion* WF-PROG-4.

PROOF. Assume

$$\texttt{implementation}\langle\overline{X}\rangle\ I\langle\overline{T}\rangle\ [\,\overline{M}\,]\ \texttt{where}\ \overline{P}\ \ldots \qquad \texttt{implementation}\langle\overline{Y}\rangle\ I\langle\overline{U}\rangle\ [\,\overline{N}\,]\ \texttt{where}\ \overline{Q}\ \ldots$$

$[\overline{V/X}]\overline{M} \trianglelefteq_c [\overline{W/Y}]\overline{N}$ and $\emptyset \Vdash [\overline{W/Y}]\overline{Q}$.

W.l.o.g., $\overline{X}\cap\overline{Y} = \emptyset$ and the two implementation definitions given are disjoint. From $[\overline{V/X}]\overline{M} \trianglelefteq_c [\overline{W/Y}]\overline{N}$ and Lemma 13.4 we get the existence of a substitution σ such that $\sigma\overline{M} \trianglelefteq_c \sigma\overline{N}$ and $[\overline{V/X}] = \sigma'\sigma$ and $[\overline{W/Y}] = \sigma'\sigma$ for some substitution σ'.

Now assume $\mathcal{P} \in [\overline{V/X}]\overline{P}$. That is, there exists some i such that $\mathcal{P} = [\overline{V/X}]P_i$. From criterion WF-PROG-4′ we then get that either $\{Q \in \sigma\overline{Q}\} \Vdash \sigma P_i$ or $\sigma P_i \in \mathsf{sup}(\sigma\overline{Q})\cup\{T\ \texttt{extends}\ U \mid T\ \texttt{extends}\ U' \in \sigma\overline{Q}, \sigma\overline{Q} \vdash_{\mathrm{q}}{}' U' \le U\}$

- Case $\{Q \in \sigma\overline{Q}\} \Vdash \sigma P_i$. We have $\emptyset \Vdash \sigma'\{Q \in \sigma\overline{Q}\}$, so $\emptyset \Vdash [\overline{V/X}]P_i$ by Lemma 7.1.
- Case $\sigma P_i \in \mathsf{sup}(\sigma\overline{Q}) \cup \{T\ \texttt{extends}\ U \mid T\ \texttt{extends}\ U' \in \sigma\overline{Q}, \sigma\overline{Q} \vdash_{\mathrm{q}}{}' U' \le U\}$.

  If $\sigma P_i \in \mathsf{sup}(\sigma\overline{Q})$, then $[\overline{V/X}]P_i \in \mathsf{sup}([\overline{W/Y}]\overline{Q})$ by Lemma 6.15. We then get with $\emptyset \Vdash [\overline{W/Y}]\overline{Q}$, Theorem 6.34, Lemma 6.28, and Theorem 6.36. that $\emptyset \Vdash [\overline{V/X}]P_i$.

  Suppose $\sigma P_i \in \{T\ \texttt{extends}\ U \mid T\ \texttt{extends}\ U' \in \sigma\overline{Q}, \sigma\overline{Q} \vdash_{\mathrm{q}}{}' U' \le U\}$ and assume $\sigma P_i = T\ \texttt{extends}\ U$ with $T\ \texttt{extends}\ U' \in \sigma\overline{Q}$ and $\sigma\overline{Q} \vdash_{\mathrm{q}}{}' U' \le U$. With $\emptyset \Vdash [\overline{W/Y}]\overline{Q}$ then

  $$\emptyset \vdash \sigma' T \le \sigma' U'$$

  and with rule SUB-Q-ALG-KERNEL, Theorem 6.36, and Lemma 7.1:

  $$\emptyset \vdash \sigma' U' \le \sigma' U$$

  By transitivity of subtyping and rule ENT-EXTENDS then $\emptyset \Vdash [\overline{V/X}]P_i$.

This proves $\emptyset \Vdash [\overline{V/X}]\overline{P}$. □

Lemma 13.8. *Criterion* WF-TENV-7 *and criterion* WF-TENV-7′ *are equivalent.*

PROOF. The proof is easy, using Lemma 13.4 for the implication "⇐". □

13.2 Finitary Closure of Types

Criterion WF-TENV-3 requires $\mathsf{cls}_\Delta(\mathscr{T})$ to be finite for every finite set of types $\mathscr{T}$. We now present an equivalent, syntactic characterization of this property, which was originally developed by [3]. Most definitions, lemmas, and proofs in this section are heavily based on work by [2].

Definition 13.9 (Type parameter dependency graph)**.** *The* type parameter dependency graph *$\mathscr{D}$ is a labeled graph $\mathscr{D} = (\mathscr{V}, \mathscr{E})$.*

The set of vertices $\mathscr{V}$ consists of all the formal type parameters to classes in the program:

$$\mathscr{V} = \{C\#i \mid \texttt{class}\ C\langle\overline{X}^n\rangle\ \texttt{extends}\ N \ldots, i \in [n]\}$$

At some points, we use the name of the formal type parameter X_i instead of $C\#i$, assuming the names of all formal type parameters are (α-converted to be) distinct.

The set of labeled edges $\mathscr{E} = \mathscr{E}_0 \cup \mathscr{E}_1$, where the labels are drawn from the set $\{0, 1\}$, represent uses of formal type parameters. Edges labeled with 0 are called non-expansive edges:

$$\mathscr{E}_0 = \{C\#i \xrightarrow{0} D\#j \mid \texttt{class}\ C\langle\overline{X}^n\rangle\ \texttt{extends}\ N \ldots, D\langle\overline{T}\rangle\ \textit{subterm of}\ N, X_i = T_j\}$$

Edges labeled with 1 are called expansive edges:

$$\mathscr{E}_1 = \{C\#i \xrightarrow{1} D\#j \mid \texttt{class}\ C\langle\overline{X}^n\rangle\ \texttt{extends}\ N \ldots, D\langle\overline{T}\rangle\ \textit{subterm of}\ N,\ X_i\ \textit{proper subterm of}\ T_j\}$$

The type parameter dependency graph is said to be expansive *iff it contains a cycle with at least one expansive edge. Otherwise, the type parameter dependency graph is said to be* non-expansive.

Definition 13.10 (Levels in the type parameter dependency graph)**.** *Let $\mathscr{D} = (\mathscr{V}, \mathscr{E})$ be a type parameter dependency graph. The* level *of a vertex $X \in \mathscr{V}$, written $\mathsf{level}(X)$, is a natural number such that for $X, Y \in \mathscr{V}$ the following property holds:*

$$\text{if } X \to Y \text{ and } Y \to^{+} X \text{ then } \mathsf{level}(X) = \mathsf{level}(Y)$$
$$\text{if } X \to Y \text{ and not } Y \to^{+} X \text{ then } \mathsf{level}(X) > \mathsf{level}(Y)$$

Definition 13.11 (Paths)**.** *A* path *is a sequence of formal type parameters, where ϵ denotes the empty path and $X.p$ is the path consisting of formal type parameter X prepended to path p. By interpreting a path p as a partial function from terms to subterms, we may use p to identify a particular subterm in a type:*

$$\epsilon(T) = T \qquad\qquad \frac{p(T_i) = U}{(C\#i.p)(C\langle\overline{T}\rangle) = U}$$

We say that p is a path in T *if $p(T)$ is defined.*

Definition 13.12. *Let $L, \delta \in \mathbb{N}$. The predicate $\phi_{L,\delta}(p)$ holds for a path p iff p can be divided into a sequence of (possibly empty) sequences of type parameters whose levels are bounded by $0, \ldots, L-1$ and whose lengths are bounded by δ. That is, $\phi_{L,\delta}(p)$ means that p has the form $\overline{X_0}\,\overline{X_1}\,\ldots\,\overline{X_{L-1}}$, such that, for all $l \in \{0, \ldots, L-1\}$, $\mathsf{level}(X) \leq l$ for all $X \in \overline{X_l}$ and $|\overline{X_l}| \leq \delta$.*

We extend $\phi_{L,\delta}$ to types by defining that $\phi_{L,\delta}(T)$ holds for a type T iff $\phi_{L,\delta}(p)$ holds for every path p in T.

Lemma 13.13. *If $\phi_{L,\delta}(T)$ then $\mathsf{height}(T) \leq \delta L$.*

Proof. Easy. □

Let $\mathscr{D} = (\mathscr{V}, \mathscr{E})$ be the type parameter dependency graph of the underlying program. We define $L \in \mathbb{N}$ as the number of levels in $\mathscr{D}$ (that is, $0 \leq \mathsf{level}(X) < L$ for any formal type parameter X). Moreover, we define $\delta \in \mathbb{N}$ as a bound on the height of the superclasses of the underlying program. That is, $\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N \ldots$ implies $\mathsf{height}(N) \leq \delta$. In the following, we write ϕ instead of $\phi_{L,\delta}$.

Lemma 13.14. *If $\mathsf{height}(T) \leq \delta$ then $\phi(T)$.*

Proof. Easy. □

Lemma 13.15. *If $N \trianglelefteq_{c} M$ and $\phi(N)$ then $\phi(M)$.*

Proof. We proceed by induction on the derivation of $N \trianglelefteq_{c} M$. If the last rule in this derivation is ext-c-refl, then the claim holds trivially. Otherwise, we have

$$\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N' \ldots$$
$$[\overline{T/X}]N' \trianglelefteq_{c} M$$
$$N = C\langle\overline{T}\rangle$$

We now show $\phi([\overline{T/X}]N')$, the claim then follows by the I.H. Note that $\mathsf{height}(N') \leq \delta$ by definition of δ.

Consider a path p in $[\overline{T/X}]N'$. There are two possibilities. First, p could be simply a path in N' that maps to a non-variable type. In this case, we know $|p| \leq \delta$, so we have $\phi(p)$ immediately.

Otherwise $p = p'.q$ for paths p' and q such that p' is non-empty, $p'(N') = X_i$ and q is a path in T_i. Hence, $C\#i.q$ is a path in $C\langle\overline{T}\rangle$, and so from $\phi(C\langle\overline{T}\rangle)$, we can deduce $\phi(C\#i.q)$, or written another way, $\phi(X_i.q)$. Now if $\mathsf{level}(X_i) = k$ then $q = \overline{Y_k}\,\overline{Y_{k+1}}\,\ldots\,\overline{Y_{L-1}}$, with $\mathsf{level}(Y_{li}) \leq l$ for all i and $k \leq l < L$ and with $|\overline{Y_k}| < \delta$ and $|\overline{Y_l}| \leq \delta$ for $k < l < L$. Suppose $p' = \overline{Z}.Z$. By definition of the type parameter dependency graph, we know that $X_i \xrightarrow{1} Z_j$ for each j and that $X_i \xrightarrow{0} Z$. The type parameter dependency graph is non-expansive, so there is no j such that $Z_j \to^{+} X_i$. Hence, $\mathsf{level}(Z_j) < \mathsf{level}(X_i) = k$ for each j. Finally, because $|\overline{Z}| < \delta$ and $\mathsf{level}(Z) \leq k$ and $|\overline{Y_k}| < \delta$, we see that $p = \overline{Z}\,(Z.\overline{Y_k})\,\overline{Y_{k+1}}\,\ldots\,\overline{Y_{L-1}}$ satisfies ϕ, as required. □

Lemma 13.16. *Assume that δ is not only a bound on the height of the superclasses of the underlying program, but also a bound on the height of the types in Δ. If $\Delta \vdash_{\mathrm{q}}{}' U \leq N$ and $\phi(U)$, then $\phi(N)$.*

PROOF. We proceed by induction on the derivation of $\Delta \vdash_{\mathrm{q}}{}' U \leq N$.
Case distinction on the last rule in the derivation of $\Delta \vdash_{\mathrm{q}}{}' U \leq N$.

- *Case* rule SUB-Q-ALG-OBJ: Trivial.
- *Case* rule SUB-Q-ALG-VAR-REFL: Impossible.
- *Case* rule SUB-Q-ALG-VAR: Then $X = U$, $X \,\texttt{extends}\, U' \in \Delta$, and $\Delta \vdash_{\mathrm{q}}{}' U' \leq N$. Hence, $\mathsf{height}(U') \leq \delta$, so $\phi(U')$ by Lemma 13.14. The claim now follows from the I.H.
- *Case* rule SUB-Q-ALG-CLASS: Follows by Lemma 13.15.
- *Case* rule SUB-Q-ALG-IFACE: Impossible.

End case distinction on the last rule in the derivation of $\Delta \vdash_{\mathrm{q}}{}' U \leq N$. □

Lemma 13.17. *Suppose Δ is finite and assume that the type parameter dependency graph is non-expansive. Then $\mathsf{cls}_\Delta(\mathscr{T})$ is finite for every finite $\mathscr{T}$.*

PROOF. Let $\mathscr{T}$ be a finite set of types. We can safely assume that δ is not only a bound on height of the superclasses of the underlying program, but also a bound on the height of the types in $\mathscr{T}$ and Δ. We now prove that the height of types in $\mathsf{cls}_\Delta(\mathscr{T})$ is bounded by δL; then, because the set of types of a certain height is finite, it follows that $\mathsf{cls}_\Delta(\mathscr{T})$ is finite.

By Lemma 13.13, it suffices to show that ϕ holds for all types in $\mathsf{cls}_\Delta(\mathscr{T})$. Assume $T \in \mathsf{cls}_\Delta(\mathscr{T})$. We proceed by induction on the derivation of $T \in \mathsf{cls}_\Delta(\mathscr{T})$.
Case distinction on the last rule of the derivation of $T \in \mathsf{cls}_\Delta(\mathscr{T})$.

- *Case* rule CLS-ID: Then T in $\mathscr{T}$, so $\mathsf{height}(T) \leq \delta$. Then $\phi(T)$ with Lemma 13.14.
- *Case* rule CLS-UP: Then we have $U \in \mathsf{cls}_\Delta(\mathscr{T})$ and $\Delta \vdash_{\mathrm{a}}{}' U \leq N$ and $T = N$. From the I.H. we get $\phi(U)$. Moreover, with Theorem 10.10 we have $\Delta \vdash_{\mathrm{q}}{}' U \leq N$. The claim now follows with Lemma 13.16.
- *Case* rule CLS-DECOMP: Then $B\langle\overline{U}\rangle \in \mathsf{cls}_\Delta(\mathscr{T})$ and $T = U_i$. From the I.H. we know $\phi(B\langle\overline{U}\rangle)$, so $\phi(U_i)$ also holds.

End case distinction on the last rule of the derivation of $T \in \mathsf{cls}_\Delta(\mathscr{T})$. □

Lemma 13.18. *Suppose $C\langle\overline{T}\rangle \in \mathsf{cls}_\Delta(\mathscr{T})$.*

(*i*) *If $C\#i \xrightarrow{0} D\#j$ then $D\langle\overline{U}\rangle \in \mathsf{cls}_\Delta(\mathscr{T})$ for some $\overline{U}$ with $U_j = T_i$.*

(*ii*) *If $C\#i \xrightarrow{1} D\#j$ then $D\langle\overline{U}\rangle \in \mathsf{cls}_\Delta(\mathscr{T})$ for some $\overline{U}$ such that T_i is a proper subterm of U_j.*

PROOF.

(i) From the definition of the type parameter dependency graph, we get

$$\begin{gathered}\texttt{class}\ C\langle\overline{X}\rangle\ \texttt{extends}\ N\ \dots\\ D\langle\overline{V}\rangle \text{ subterm of } N\\ V_j = X_i\end{gathered}$$

Obviously, $\Delta \vdash_{\mathrm{a}}{}' C\langle\overline{T}\rangle \leq [\overline{T/X}]N$, so we have with rule CLS-UP that

$$[\overline{T/X}]N \in \mathsf{cls}_\Delta(\mathscr{T})$$

Possibly repeated applications of rule CLS-DECOMP then yield $[\overline{T/X}]D\langle\overline{V}\rangle \in \mathsf{cls}_\Delta(\mathscr{T})$, from which the claim follows immediately.

(ii) Similar. ☐

Lemma 13.19. *Assume* $\mathsf{cls}_\Delta(\mathscr{T})$ *is finite for every finite* $\mathscr{T}$*. Then the type parameter dependency graph is non-expansive.*

Proof. We prove the contraposition; that is, we assume that the type parameter dependency graph is expansive and show that there exists a finite set $\mathscr{T}$ such that $\mathsf{cls}_\Delta(\mathscr{T})$ infinite.

Suppose the type parameter dependency graph is expansive; that is, there is a cycle such that at least one of the edges of the cycle (say the first) is expansive. Thus, either $C\#i \xrightarrow{1} C\#i$ or $C\#i \xrightarrow{1} D\#j \to^{+} C\#i$. Now consider $\mathscr{C} = \mathsf{cls}_\Delta(\{C\langle\overline{\mathtt{Object}}\rangle\})$.

- By possibly repeated applications of Lemma 13.18 we see that also $C\langle\overline{U_1}\rangle \in \mathscr{C}$ such that `Object` is a proper subterm of U_{1i}.
- By possibly repeated applications of Lemma 13.18 we see that also $C\langle\overline{U_2}\rangle \in \mathscr{C}$ such that U_{1i} is a proper subterm of U_{2i}.
- By possibly repeated applications of Lemma 13.18 we see that also $C\langle\overline{U_3}\rangle \in \mathscr{C}$ such that U_{2i} is a proper subterm of U_{3i}.
- ...

Hence, there is a chain of types $C\langle\overline{\mathtt{Object}}\rangle = C\langle\overline{U_0}\rangle, C\langle\overline{U_1}\rangle, C\langle\overline{U_2}\rangle, \ldots$ such that $C\langle\overline{U_i}\rangle \in \mathscr{C}$ and $C\langle\overline{U_{i+1}}\rangle$ is strictly larger than $C\langle\overline{U_i}\rangle$ for all $i \in \mathbb{N}$. Thus, $\mathscr{C}$ is infinite. ☐

We are now ready to give an equivalent formulation of criterion Wf-Tenv-3.

Wf-Tenv-3′ The type parameter dependency graph of the underlying program is non-expansive.

Lemma 13.20. *Criterion* Wf-Tenv-3 *and criterion* Wf-Tenv-3′ *are equivalent.*

Proof. Follows from Lemma 13.17, Lemma 13.19, and criterion Wf-Tenv-1. ☐

References

[1] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[2] A. J. Kennedy and B. C. Pierce. On decidability of nominal subtyping with variance. In *FOOL/WOOD, informal proceedings*, Jan. 2007. `http://foolwood07.cs.uchicago.edu/program/kennedy-abstract.html`.

[3] M. Viroli. On the recursive generation of parametric types. Technical Report DEIS-LIA-00-002, Università di Bologna, 2000.