4.4 Formulating SMT Statements

So far we have described the inference rules and the subtyping rule. We have yet to describe an algorithm that can derive a valid type for a set of given subtyping rules.

Definition 4.1: Liquid Type Variable

We say $\mathcal{K} := \{\kappa_i \mid i \in \mathbb{N}\}$ is the set of all *liquid type variables.*

Note that κ is a special character.

Definition 4.2: Template

We say $\hat{T}$ is a *template* $:\Leftrightarrow$

$$\begin{aligned}
&\hat{T} \text{ is of form } \{\nu : Int \mid [k]_S\} \\
&\quad \text{where } k \in \mathcal{K} \text{ and } S : \mathcal{V} \twoheadrightarrow \mathcal{Q} \\
\vee\ &\hat{T} \text{ is of form } a : \{\nu : Int \mid [k]_S\} \rightarrow \hat{T} \\
&\quad \text{where } k \in \mathcal{K}, \hat{T} \text{ is a template and } S : \mathcal{V} \twoheadrightarrow \mathcal{Q}.
\end{aligned}$$

We define $\mathcal{T}^? := \{\hat{T} \mid \hat{T} \text{ is a template}\}$

A template will be used for a liquid type with unknown refinement. Note that the inference rule for function applications introduces a refinement substitution S. For templates this substitution is not defined and needs to be delayed until the corresponding liquid type has been derived. We will point out whenever the substitution $[k]_S$ will be applied.

Definition 4.3: Type Variable Context

Let $\quad K := \{[k]_S \mid k \in \mathcal{K} \wedge S : \mathcal{V} \twoheadrightarrow \mathcal{Q}\}$.

We say $\Theta : \mathcal{V} \twoheadrightarrow (\mathcal{Q} \cup K)$ is a type variable context.

Our algorithm will resolve a set of suptyping conditions:

Definition 4.4: Subtyping Condition

We say c is a *Subtyping Condition* $:\Leftrightarrow$

c is of form $\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2$
where $\hat{T}_1, \hat{T}_2$ are a liquid types or templates, Θ is a type variable context and $\Lambda \subset \mathcal{Q}$.

We define $\mathcal{C} := \{c \mid c \text{ is a subtyping condition}\}$

We will also need a function to obtain the set of all liquid type variables of a template or subtyping condition.

Definition 4.5: Vars

$$\text{Vars} : (\mathcal{T} \cup \mathcal{T}^?) \rightarrow \mathcal{P}(\mathcal{K})$$
$$\text{Vars}(\{\nu \in Int|r\}) = \{\}$$
$$\text{Vars}(\{\nu \in Int|\kappa_i\}) = \{\kappa_i\}$$
$$\text{Vars}(a : \{\nu \in Int|\kappa_i\} \rightarrow \hat{T}) = \{\kappa_i\} \cup \text{Vars}(\hat{T})$$

$$\text{Vars} : \mathcal{C} \rightarrow \mathcal{P}(\mathcal{K})$$
$$\text{Vars}(\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2) = \text{Vars}(\hat{T}_1) \cup \text{Vars}(\hat{T}_2)$$
$$\cup \{k|(\_,q) \in \Theta \wedge q = [k]_S \text{ for } k \in \mathcal{K} \text{ and } S : \mathcal{V} \nrightarrow \mathcal{Q}\}$$

The main idea of the algorithm is to first generate a set of predicates and then exclude elements until all subtyping conditions are valid for the remaining predicates. By conjunction over all remaining predicates we result in a valid refinement.

We therefore need a function, depending on a set of variable Q, that will generate a set of predicates. Note that the resulting set should be finite and a subset of $\mathcal{Q}$. If the generated set is too small, then our resulting subtyping conditions might be too weak.

$$\begin{aligned}
Init : \mathcal{P}(\mathcal{V}) \rightarrow &\mathcal{P}(\mathcal{Q}) \\
Init(V) ::= &\{0 < \nu\} \\
&\cup \{a < \nu \mid a \in V\} \\
&\cup \{\nu < 0\} \\
&\cup \{\nu < a \mid a \in V\} \\
&\cup \{\nu = a \mid a \in V\} \\
&\cup \{\nu = 0\} \\
&\cup \{a < \nu \vee \nu = a \mid a \in V\} \\
&\cup \{\nu < a \vee \nu = a \mid a \in V\} \\
&\cup \{0 < \nu \vee \nu = 0\} \\
&\cup \{\nu < 0 \vee \nu = 0\} \\
&\cup \{\neg(\nu = a) \mid a \in V\} \\
&\cup \{\neg(\nu = 0)\}
\end{aligned}$$

We can always extend the realm of predicates if the resulting refinements are too weak.

4.4.1 The Inference Algorithm

$\text{Infer} : \mathcal{P}(\mathcal{C}) \to (\mathcal{K} \twoheadrightarrow \mathcal{Q})$

$\text{Infer}(C) =$

$\quad$ Let $V := \bigcup\limits_{\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2 \in C} \{a \mid (a, \_) \in \Theta\}$

$\quad\quad Q_0 := Init(V),$

$\quad\quad A_0 := \{(\kappa, Q_0) \mid \kappa \in \bigcup\limits_{c \in C} \text{Var}(c)\},$

$\quad\quad A := \text{Solve}(\bigcup\limits_{c \in C} \text{Split}(c), A_0)$

$\quad$ in $\{(\kappa, \bigwedge Q) \mid (\kappa, Q) \in A\}$

$\quad$ where $V \subseteq \mathcal{V}, Q_0, Q \subseteq \mathcal{Q}, A_0, A \in \mathcal{K} \twoheadrightarrow \mathcal{Q}, \Theta$ be a type variable context and $\Lambda \subseteq \mathcal{Q}$.

We first split the subtyping conditions for functions into subtyping conditions for simpler templates:

$$
\begin{aligned}
\mathcal{C}^- := \{ \ & \{\nu : Int|q_1\} <:_{\Theta,\Lambda} \{\nu : Int|q_2\} \\
& \mid (q_1 \in \mathcal{Q} \vee q_1 = [k_1]_{S_1} \text{ for } k_1 \in \mathcal{K}, S_1 \in \mathcal{V} \twoheadrightarrow \mathcal{Q}) \\
& \wedge (q_2 \in \mathcal{Q} \vee q_2 = [k_2]_{S_2} \text{ for } k_2 \in \mathcal{K}, S_2 \in \mathcal{V} \twoheadrightarrow \mathcal{Q})\}.
\end{aligned}
$$

With this we can now define the Split function.

$$
\begin{aligned}
&\text{Split} : \mathcal{C} \twoheadrightarrow \mathcal{P}(\mathcal{C}^-) \\
&\text{Split}(a : \{\nu : Int|q_1\} \to \hat{T}_2 <:_{\Theta,\Lambda} a : \{\nu : Int|q_3\} \to \hat{T}_4) = \\
&\quad\quad \{\{\nu : Int|q_3\} <:_{\Theta,\Lambda} \{\nu : Int|q_1\}\} \cup \text{Split}(\hat{T}_2 <:_{\Theta\cup\{(a,q_3)\},\Lambda} \hat{T}_4) \\
&\text{Split}(\{\nu : Int|q_1\} <:_{\Theta,\Lambda} \{\nu : Int|q_2\}) = \\
&\quad\quad \{\{\nu : Int|q_1\} <:_{\Theta,\Lambda} \{\nu : Int|q_2\}\}
\end{aligned}
$$

Note that Split will result in an error, if the subtyping condition is not one of the two cases above.

We resolve the obtained subtyping conditions by repeatably checking if a subtyping condition is not valid and removing all predicates that contradict it. By removing the predicate we weaken the resulting refinement.

$\text{Solve} : \mathcal{P}(\mathcal{C}^-) \times (\mathcal{K} \nrightarrow \mathcal{Q}) \to (\mathcal{K} \nrightarrow \mathcal{Q})$

$\text{Solve}(C, A) =$

$\qquad$ Let $S := \{(k, \bigwedge Q) \mid (k, Q) \in A\}$.

$\qquad$ If there exists $(\{v : Int \mid q_1\} <:_{\Theta, \Lambda} \{v : Int \mid [k_2]_{S_2}\}) \in C$ such that

$\qquad\qquad \neg(\forall i_1 \in \text{value}_\Gamma(\{\nu : Int | r_1'\}). \ldots. \forall i_n \in \text{value}_\Gamma(\{\nu : Int | r_n'\}).$

$\qquad\qquad [\![r_1 \wedge p]\!]_{\{(b_1, i_1), \ldots, (b_n, i_n)\}} \Rightarrow [\![r_2]\!]_{\{(b_1, i_1), \ldots, (b_n, i_n)\}})$

$\qquad\qquad$ for $\ r_2 := \bigwedge [S(\kappa_2)]_{S_2}, \ p := \bigwedge \Lambda,$

$$r_1 := \begin{cases} \bigwedge [S(k_1)]_{S_1} & \text{if } q_1 \text{ has the form } [k_1]_{S_1} \text{ for } k \in \mathcal{K} \text{ and } S_1 \in \mathcal{V} \nrightarrow \mathcal{Q} \\ q_1 & \text{if } q_1 \in \mathcal{Q} \end{cases},$$

$\qquad\qquad \Theta' := \{ \ (a, r)$

$\qquad\qquad\qquad\qquad | \ r \text{ has the form } q \wedge (a, q) \in \Theta \wedge q \in \mathcal{Q}$

$\qquad\qquad\qquad\qquad \vee \ r \text{ has the form } [[k]_S]_{S_0} \wedge (a, q) \in \Theta$

$\qquad\qquad\qquad\qquad\qquad \wedge q \text{ has the form } [k]_{S_0} \wedge k \in \mathcal{K} \wedge S_0 \in \mathcal{V} \nrightarrow \mathcal{Q}\}$

$\qquad\qquad \{(b_1, r_1'), \ldots, (b_n, r_n')\} = \Theta'$

$\qquad$ then $\text{Solve}(C, \text{Weaken}(c, A))$ else A

$\qquad$ where $k, k_2 \in \mathcal{K}, S : \mathcal{K} \nrightarrow \mathcal{Q}, Q, \Lambda \subseteq \mathcal{Q}, S_1, S_2 : \mathcal{V} \nrightarrow \mathcal{Q}, q_1 \in \mathcal{K} \cup \mathcal{Q},$

$\qquad$ Θ be a type variable context, $r_1, p, r_2 \in \mathcal{Q}, a \in \mathcal{V}, \Theta' : \mathcal{V} \nrightarrow \mathcal{Q}, r \in \mathcal{Q}, n \in \mathbb{N}, b_i \in \mathcal{V},$

$\qquad$ $r_i \in \mathcal{Q}$ for $i \in \mathbb{N}_0^n$ and $[t]_A$ denotes the substitution for the term t with a

$\qquad$ substitution A.

Note that we can use a SMT solver to validate

$$\neg(\forall i_1 \in \text{value}_\Gamma(\{\nu : Int | r_1'\}). \ldots. \forall i_n \in \text{value}_\Gamma(\{\nu : Int | r_n'\}).$$

$$[\![r_1 \wedge p]\!]_{\{(b_1, i_1), \ldots, (b_n, i_n)\}} \Rightarrow [\![r_2]\!]_{\{(b_1, i_1), \ldots, (b_n, i_n)\}})$$

by deciding the satisfiablity of

$$(\bigwedge_{j=0}^{n} r_j') \wedge r_1 \wedge p \wedge \neg r_2$$

with free variables $b_i \in \mathbb{N}$ for $i \in \mathbb{N}_1^n$.

Weaken $: \mathcal{C}^- \times (\mathcal{K} \nrightarrow \mathcal{Q}) \nrightarrow (\mathcal{K} \nrightarrow \mathcal{Q})$

Weaken$(\{\nu : Int|x\} <:_{\Theta,\Lambda} \{\nu : Int|[k_2]_{S_2}\}, A) =$

   Let $S := \{(k, \bigwedge Q) \mid (k, Q) \in A\}$,

$$r_1 := \begin{cases} \bigwedge [S(k_1)]_{S_1} & \text{if } q_1 \text{ has the form } [k_1]_{S_1} \text{ for } k \in \mathcal{K} \text{ and } S_1 \in \mathcal{V} \nrightarrow \mathcal{Q} \\ q_1 & \text{if } q_1 \in \mathcal{Q} \end{cases},$$

   $p := \bigwedge \{[q]_S \mid q \in \Lambda\}$,

   $\Theta' := \{ \, (a, r)$

          $\mid r$ has the form $q \wedge (a, q) \in \Theta \wedge q \in \mathcal{Q}$

          $\vee\ r$ has the form $[[k]_S]_{S_0} \wedge (a, q) \in \Theta$

               $\wedge\ q$ has the form $[k]_{S_0} \wedge k \in \mathcal{K} \wedge S_0 \in \mathcal{V} \nrightarrow \mathcal{Q}\}$

   $\{(b_1, r'_1), \dots, (b_n, r'_n)\} = \Theta'$

   $Q_2 := \{ \, q$

          $\mid q \in A(k_2) \wedge \text{wellFormed}(q, \{(b_1, \{\nu : Int|r'_1\}), \dots, (b_n, \{\nu : Int|r'_n\})\})$

          $\wedge (\forall i_1 \in \text{value}_\Gamma(\{\nu : Int|r'_1\}). \dots. \forall i_n \in \text{value}_\Gamma(\{\nu : Int|r'_n\}).$

               $[\![r_1 \wedge p]\!]_{\{(b_1,i_1),\dots,(b_n,i_n)\}} \Rightarrow [\![q]_{S_2}]\!]_{\{(b_1,i_1),\dots,(b_n,i_n)\}})\}$

   in $\{(k, Q) \mid (k, Q) \in A \wedge k \neq k_2\} \cup \{(k_2, Q_2)\}$

   where $k, k_1 \in \mathcal{K}, Q, Q_2 \subseteq \mathcal{Q}, S : \mathcal{K} \nrightarrow \mathcal{Q}, r_1 \in \mathcal{Q}, p \in \mathcal{Q}, S_1, S_2 : \mathcal{V} \nrightarrow \mathcal{Q}, \Theta' : \mathcal{V} \nrightarrow \mathcal{T},$
   $a \in \mathcal{V}, T' \in \mathcal{T} \cup \mathcal{T}^? n \in \mathbb{N}, b_i \in \mathcal{V}, T_i \in \mathcal{T}$ for $i \in \mathbb{N}_0^n$ and $[t]_A$ denotes the
   substitution for the term t with a substitution A.

Note that we can use a SMT solver to validate

$$\forall i_1 \in \text{value}_\Gamma(\{\nu : Int|r'_1\}). \dots. \forall i_n \in \text{value}_\Gamma(\{\nu : Int|r'_n\}).$$
$$[\![r_1 \wedge p]\!]_{\{(b_1,i_1),\dots,(b_n,i_n)\}} \Rightarrow [\![q]_{S_2}]\!]_{\{(b_1,i_1),\dots,(b_n,i_n)\}}$$

To do so, we first need to compute $r_2 := [q]_{S_2}$, with that we can now use a SMT
solver to decide the satisfiablity of

$$\neg((\bigwedge_{j=0}^n r'_j) \wedge r_1 \wedge p) \vee r_2$$

with free variables $b_i \in \mathbb{N}$ for $i \in \mathbb{N}_1^n$.

> **Example 4.1**
>
>   Assume that we have given the following suptyping conditions:
>
> $\Theta := \{(a, \{Int|\kappa_1\}), (b, \{Int|\kappa_2\})\}$
> $C_0 := \{\{\nu : Int|\nu = b\} <:_{\Theta,\{a<b\}} \{\nu : Int|\kappa_3\},$
> $\{\nu : Int|\nu = a\} <:_{\Theta,\{\neg(a<b)\}} \{\nu : Int|\kappa_3\},$
> $a : \{\nu : Int|\kappa_1\} \to b : \{\nu : Int|\kappa_2\} \to \{\nu : Int|\kappa_3\}$
> $<:_{\{\},\{\}} a : \{\nu : Int|\, True\} \to b : \{\nu : Int|\, True\} \to \{\nu : Int|\kappa_4\}$

Then $V := \{a, b\}$ and $A_0 := \{(\kappa_1, Init(V)), (\kappa_2, Init(V)), (\kappa_3, Init(V)), (\kappa_4, Init(V))\}$.

Splitting the conditions

We will only consider the last subtyping condition of C_0, all other conditions do not need to be split.

$$
\begin{aligned}
&\text{Split}(a : \{\nu : Int|\kappa_1\} \to b : \{\nu : Int|\kappa_2\} \to \{\nu : Int|\kappa_3\} \\
&\qquad <:_{\{\},\{\}} a : \{\nu : Int|\,True\} \to b : \{\nu : Int|\,True\} \to \{\nu : Int|\kappa_4\}) \\
&= \text{Split}(a : \{\nu : Int|\kappa_1\} <:_{\{\},\{\}} a : \{\nu : Int|\,True\}) \\
&\quad \cup \text{Split}(b : \{\nu : Int|\kappa_2\} \to \{\nu : Int|\kappa_3\} \\
&\qquad <:_{\{(a,\{\nu:Int|\,True\})\},\{\}} b : \{\nu : Int|\,True\} \to \{\nu : Int|\kappa_4\}) \\
&= \{a : \{\nu : Int|\,True\} <:_{\{\},\{\}} a : \{\nu : Int|\kappa_1\}\} \\
&\quad \cup \text{Split}(b : \{\nu : Int|\,True\} <:_{\{(a,\{\nu:Int|\,True\})\},\{\}} b : \{\nu : Int|\kappa_2\}) \\
&\quad \cup \text{Split}(\{\nu : Int|\kappa_3\} <:_{\Theta,\{\}} \{\nu : Int|\kappa_4\}) \\
&= \{\{\nu : Int|\,True\} <:_{\{\},\{\}} \{\nu : Int|\kappa_1\}, \\
&\quad \{\nu : Int|\,True\} <:_{\{(a,\{\nu:Int|\,True\})\},\{\}} \{\nu : Int|\kappa_2\}, \\
&\quad \{\nu : Int|\kappa_3\} <:_{\{\Theta\},\{\}} \{\nu : Int|\kappa_4\}\}
\end{aligned}
$$

So in conclusion we have the following set of subtypings conditions:

$$
\begin{aligned}
C := \{&\{\nu : Int|\nu = b\} <:_{\Theta,\{a<b\}} \{\nu : Int|\kappa_3\}, \\
&\{\nu : Int|\nu = a\} <:_{\Theta,\{\neg(a<b)\}} \{\nu : Int|\kappa_3\}, \\
&\{\nu : Int|\,True\} <:_{\{\},\{\}} \{\nu : Int|\kappa_1\}, \\
&\{\nu : Int|\,True\} <:_{\{(a,\{\nu:Int|\,True\})\},\{\}} \{\nu : Int|\kappa_2\}, \\
&\{\nu : Int|\kappa_3\} <:_{\Theta,\{\}} \{\nu : Int|\kappa_4\}\}
\end{aligned}
$$

We therefore now will go through each condition $c \in C$ and check its validity.

Iteration 1, Case $c = \{\nu : Int|\nu = b\} <:_{\Theta,\{a<b\}} \{\nu : Int|\kappa_3\}$:

We define $S := \{(\kappa_1, \bigwedge Init(V)), (\kappa_2, \bigwedge Init(V)), (\kappa_3, \bigwedge Init(V)), (\kappa_4, \bigwedge Init(V))\}$.

$Init(V)$ contains $\nu = 0$ and $\neg \nu = 0$, so we know that $\bigwedge Init(V)$ can be simplified to *False*.

We now check if

$$\forall a \in \text{values}_{\{\}}(\{\nu : Int | False\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int | False\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\nu = b \wedge a < b$$
$$\models \forall a \in \text{values}_{\{\}}(\{\nu : Int | False\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int | False\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$False$$

is not valid.

We know that $\text{values}_{\{\}}(\{\nu : False\}) = \{\}$, and therefore this can be simplified to $True \models True$, which is valid.

Iteration 1, Case $c = \{\nu : Int | \nu = a\} <:_{\Theta, \{\neg(a<b)\}} \{\nu : Int | \kappa_3\}$:

The argument is analogously to the previous case.

Iteration 1, Case $c = \{\nu : Int | True\} <:_{\{\}, \{\}} \{\nu : Int | \kappa_1\}$:

We now check if

$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}). True \models \forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}). False$$

is valid. This time we can ignore the quantifiers and thus it simplifies to $True \models False$, which is not valid.

We therefore will now weaken A_0. To do so we compute all $q \in A_0(\kappa_1)$ such that wellFormed(q) and

$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}).[\![True]\!]_{\{\}} \models \forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}).[\![q]\!]_{\{\}}.$$

There are only two values for q that are well formed: *True* and *False*.

The resulting set is $Q_2 := \{True\}$ and thus we replace A_0 with

$$A := \{(\kappa_1, \{True\}), (\kappa_2, Init(V)), (\kappa_3, Init(V)), (\kappa_4, Init(V))\}$$

Iteration 1, Case $c = \{\nu : Int | True\} <:_{\{(a, \{\nu:Int|True\})\}, \{\}} \{\nu : Int | \kappa_2\}$:

The argument is analogously to the previous case, resulting in the updated value for A:

$$A = \{(\kappa_1, \{True\}), (\kappa_2, \{True\}), (\kappa_3, Init(V)), (\kappa_4, Init(V))\}$$

Iteration 1, Case $c = \{\nu : Int | \kappa_3\} <:_{\Theta, \{\}} \{\nu : Int | \kappa_4\}\}$:

The suptyping condition is valid, analogously to the first case of this iteration.

Iteration 2, Case $c = \{\nu : Int | \nu = b\} <:_{\Theta, \{a < b\}} \{\nu : Int | \kappa_3\}$:

We check the validity of

$$\forall a \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\nu = b \wedge a < b$$
$$\models \forall a \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$False.$$

It is easy to see, that it is not valid.

Thus we now compute all $q \in A(\kappa_3)$ such that wellFormed(q) and

$$\forall a \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\nu = b \wedge a < b$$
$$\models \forall a \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | True\}).$$
$$q.$$

is valid. The resulting set Q_2 is the following.

$$Q_2 := \{a < \nu, \nu = b, \neg(\nu = a), \nu < b \vee \nu = b, b < \nu \vee \nu = b, \nu < a \vee \nu = a,$$
$$a < \nu \vee \nu = a\}$$

Therefore we update A:

$$A = \{(\kappa_1, \{True\}), (\kappa_2, \{True\}),$$
$$(\kappa_3, \{a < \nu, \nu = b, \neg(\nu = a), \nu < b \vee \nu = b, b < \nu \vee \nu = b, \nu < a \vee \nu = a,$$
$$a < \nu \vee \nu = a\}),$$
$$(\kappa_4, Init(V))\}$$

Iteration 2, Case $c = \{\nu : Int | \nu = a\} <:_{\Theta, \{\neg(a < b)\}} \{\nu : Int | \kappa_3\}$:

We check the validity of

$$\forall a \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\nu = a \wedge \neg(a < b)$$
$$\vDash \forall a \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$a < \nu \wedge \nu = b \wedge \neg(\nu = a) \wedge \nu < b \vee \nu = b$$
$$\wedge\, b < \nu \vee \nu = b \wedge \nu < a \vee \nu = a \wedge a < \nu \vee \nu = a.$$

It is not valid, because $\nu = a \wedge \neg(a < b) \vDash \nu = b$ is not valid.

Thus we compute all $q \in A(\kappa_3)$ such that wellFromed(q) and

$$\forall a \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\nu = a \wedge \neg(a < b)$$
$$\vDash \forall a \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int \,|\, True\}).$$
$$q.$$

is valid. The resulting set Q_2 is the following.

$$Q_2 := \{\nu < b \vee \nu = b, \nu < a \vee \nu = a\}$$

Thus we update A:

$$A = \{(\kappa_1, \{True\}), (\kappa_2, \{True\}), \quad (\kappa_3, \{\nu < b \vee \nu = b, \nu < a \vee \nu = a\}), (\kappa_4, Init(V))\}$$

Iteration 2, Case $\{\nu : Int \,|\, True\} <:_{\{\},\{\}} \{\nu : Int \,|\, \kappa_1\}$:

Nothing has changed since the last iteration, therefore this case can be skipped.

Iteration 2, Case $\{\nu : Int \,|\, True\} <:_{\{(a, \{\nu : Int \,|\, True\})\},\{\}} \{\nu : Int \,|\, \kappa_2\}$:

The argument is analogously to the previous case, therefore this case can be skipped.

Iteration 2, Case : $\{\nu : Int \,|\, \kappa_3\} <:_{\Theta,\{\}} \{\nu : Int \,|\, \kappa_4\}$:

We check the validity of

$$\forall a \in \text{values}_{\{\}}(\{\nu : Int | \mathit{True}\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int | \mathit{True}\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | \mathit{True}\}).$$
$$\{\nu < b \vee \nu = b \wedge \nu < a \vee \nu = a\}$$
$$\vDash \forall a \in \text{values}_{\{\}}(\{\nu : Int | \mathit{True}\}).$$
$$\forall b \in \text{values}_{\{\}}(\{\nu : Int | \mathit{True}\}).$$
$$\forall \nu \in \text{values}_{\{\}}(\{\nu : Int | \mathit{True}\}).$$
$$\mathit{False}.$$

We see that this is not valid, therefore we derive the new set Q_2. Note that $A(\kappa_3) \subseteq Init(V)$ and therefore $Q_2 = A(\kappa_3)$.

We update the corresponding entry in A:

$$A = \{(\kappa_1, \{\mathit{True}\}), (\kappa_2, \{\mathit{True}\}),$$
$$(\kappa_3, \{\nu < b \vee \nu = b, \nu < a \vee \nu = a\}),$$
$$(\kappa_4, \{\nu < b \vee \nu = b, \nu < a \vee \nu = a\})\}$$

Iteration 3:

In this iteration all subtyping conditions are valid, thus the algorithm stops. The resulting set of substitutions is therefore the following

$$\{(\kappa_1, \mathit{True}), (\kappa_2, \mathit{True}),$$
$$(\kappa_3, (\nu < b \vee \nu = b) \wedge (\nu < a \vee \nu = a)),$$
$$(\kappa_4, (\nu < b \vee \nu = b) \wedge (\nu < a \vee \nu = a))\}$$

4.4.2 Correctness

The algorithm that we described can fail if the subtyping conditions are not well-formed.

Definition 4.6: Wellformed Subtyping Condition

We say a subtyping condition c is well-formed if the following holds

$$c \text{ has the form } \{\nu : Int | [k_1]_{S_1}\} <:_{\Theta, \Lambda} \{\nu : Int | [k_2]_{S_2}\}$$
$$\vee\ c \text{ has the form } \{\nu : Int | r\} <:_{\Theta, \Lambda} \{\nu : Int | [k_2]_{S_2}\}$$
$$\vee\ c \text{ has the form } \{\nu : Int | [k_1]_{S_1}\} \to \hat{T}_1 <:_{\Theta, \Lambda} \{\nu : Int | r\} \to \hat{T}_2$$
$$\text{such that } \hat{T}_1 <:_{\Theta, \Lambda} \hat{T}_2 \text{ is well-formed.}$$

where $r \in \mathcal{Q}$, $k_1, k_2 \in \mathcal{K}$, $S_1, S_2 : \mathcal{V} \nrightarrow \mathcal{Q}$, Θ is a type variable context, $\Lambda \subset \mathcal{Q}$ and $T_1, T_2 \in (\mathcal{T} \cup \mathcal{T}^?)$

Theorem 4.1

Let C be a set of well-formed conditions, $S := \text{Infer}(C)$ and $V := \bigcup_{\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2 \in C} \{a \mid (a, \_) \in \Theta\}$.

___

For every subtyping condition $(\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2) \in C$, let

$$\Theta' := \{ (a, r)$$
$$\mid r \text{ has the form } q \wedge (a, q) \in \Theta \wedge q \in \mathcal{Q}$$
$$\vee\, r \text{ has the form } [[k]_S]_{S_0} \wedge (a, q) \in \Theta$$
$$\wedge\, q \text{ has the form } [k]_{S_0} \wedge k \in \mathcal{K} \wedge S_0 \in \mathcal{V} \twoheadrightarrow \mathcal{Q}\}$$

and $\{(b_1, r'_1), \ldots, (b_n, r'_n)\} := \Theta'$. Then we have the following correctness property:

$[\hat{T}_1]_S \in \mathcal{T} \wedge [\hat{T}_2]_S \in \mathcal{T}$
$\wedge\ [\hat{T}_1]_S <:_{\Theta',\Lambda} [\hat{T}_2]_S$
$\wedge\ \forall S' \in (\mathcal{V} \to \mathcal{Q}).(\forall a \in \mathcal{V}.S'(a) \text{ is well defined} \Rightarrow \exists Q \in \textit{Init}(V).S'(a) = \bigwedge Q)$
$\quad \wedge\, [\hat{T}_1]_{S'} \in \mathcal{T} \wedge [\hat{T}_2]_{S'} \in \mathcal{T}$
$\quad \wedge\, ([\hat{T}_1]_{S'} <:_{\Theta',\Lambda} [\hat{T}_2]_{S'}$
$\qquad \Rightarrow (\forall a \in \mathcal{V}.S(a) \text{ and } S'(a) \text{ are well defined}$
$\qquad\quad \Rightarrow (\forall i_1 \in \text{value}_\Gamma(\{\nu : Int|r'_1\}).\ldots \forall i_n \in \text{value}_\Gamma(\{\nu : Int|r'_n\}).$
$\qquad\quad [\![S(a)]\!]_{(b_1,i_1),\ldots,(b_n,i_n)} \Rightarrow [\![S'(a)]\!]_{(b_1,i_1),\ldots,(b_n,i_n)})))$

The first post-condition states that $[\hat{T}_1]_S$ and $[\hat{T}_2]_S$ are not templates. The second condition states that S is a solution, meaning that $[\hat{T}_1]_S$ is a subtype of $[\hat{T}_2]_S$. The third condition states that S is the sharpest solution.

Proof. Let C be a set of well-formed conditions, $S := \text{Infer}(C)$ and $V := \bigcup_{\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2 \in C} \{a \mid (a, \_) \in \Theta\}$. Let $Q_0 := \textit{Init}(V)$, $A_0 := \{(\kappa, Q_0) \mid \kappa \in \bigcup_{c \in C} \text{Var}(c)\}$ and $C_1 := \bigcup_{c \in C} \text{Split}(c)$. Then by Theorem 4.2 $C_1 \subset \mathcal{C}^-$ is a set of well-formed conditions. Let $A := \text{Solve}(C_1, A_0)$ and therefore $S = \{(\kappa, \bigwedge Q) \mid (\kappa, Q) \in A\}$. Then by Theorem 4.3 the conclusion holds. $\qquad\square$

Theorem 4.2

Let c is a well-formed condition and $C := \text{Split}(c)$.

___

Then $C \subseteq \mathcal{C}^-$ and, for every $c \in C$, c is a well-formed condition.

Proof. Let c is a well-formed condition. Then $C := \text{Split}(c)$ is well-defined. We prove the theorem by induction on well-formed conditions.

Case $c = \{\nu : Int|q\} <:_{\Theta,\Lambda} \{\nu : Int|[k_2]_{S_2}\}$: c is well-formed and $c \in C^-$. Then $C = c$ and therefore the conclusion holds.

Case $c = \{\nu : Int|[k_1]_{S_1}\} \to \hat{T}_1 <:_{\Theta,\Lambda} \{\nu : Int|r\} \to \hat{T}_2$ such that $\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2$ is well-formed: Let $C_0 := \mathrm{Split}(\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2)$. By the induction hypothesis $C_0 \subseteq \mathcal{C}^-$ and for every $c \in C_0$, c is a well-formed condition. Therefore, by appling Theorem **??** to each element in $C := \{\{\nu : Int|r\} <:_{\Theta,\Lambda} \{\nu : Int|[k_1]_{S_1}\}\} \cup C_0$, the conlusion holds. $\qquad\square$

> **Theorem 4.3**
>
> Let $C \subseteq \mathcal{C}^-$ be a set of well-formed conditions, $A_1, A_2 : \mathcal{K} \nrightarrow \mathcal{Q}$ and $V := \bigcup_{\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2 \in C} \{a \mid (a, \_) \in \Theta\}$. Let for all $a \in V$, $A_1(a)$ be well defined. Let $A_2 = \mathrm{Solve}(C, A_1)$ and $S = \{(\kappa, \bigwedge Q) \mid (\kappa, Q) \in A_2\}$.
>
> ---
>
> Then for every $a \in V$, $A_2(a) \subseteq A_1(a)$
>
> For every subtyping condition $(\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2) \in C$, let
>
> $$\Theta' := \{\ (a, r)$$
> $$\mid r \text{ has the form } q \wedge (a, q) \in \Theta \wedge q \in \mathcal{Q}$$
> $$\vee\ r \text{ has the form } [[k]_S]_{S_0} \wedge (a, q) \in \Theta$$
> $$\wedge\ q \text{ has the form } [k]_{S_0} \wedge k \in \mathcal{K} \wedge S_0 \in \mathcal{V} \nrightarrow \mathcal{Q}\}$$
>
> and $\{(b_1, r_1'), \ldots, (b_n, r_n')\} = \Theta'$. We then have the following correctness property.
>
> $$[\hat{T}_1]_S \in \mathcal{T} \wedge [\hat{T}_2]_S \in \mathcal{T}$$
> $$\wedge\ [\hat{T}_1]_S <:_{\Theta',\Lambda} [\hat{T}_2]_S$$
> $$\wedge\ \forall S' \in (\mathcal{V} \to \mathcal{Q}).(\forall a \in V.\exists Q \in \mathcal{P}(A_1(a)).S'(a) = \bigwedge Q)$$
> $$\wedge\ [\hat{T}_1]_{S'} \in \mathcal{T} \wedge [\hat{T}_2]_{S'} \in \mathcal{T}$$
> $$\wedge\ ([\hat{T}_1]_{S'} <:_{\Theta',\Lambda} [\hat{T}_2]_{S'}$$
> $$\Rightarrow \forall a \in V.\forall i_1 \in \mathrm{value}_\Gamma(\{\nu : Int|r_1'\}).\ldots.\forall i_n \in \mathrm{value}_\Gamma(\{\nu : Int|r_n'\}).$$
> $$[\![S(a)]\!]_{(b_1,i_1),\ldots,(b_n,i_n)} \Rightarrow [\![S'(a)]\!]_{(b_1,i_1),\ldots,(b_n,i_n)})$$

The post-condition is the same as for Infer.

Proof. Let $C \subseteq \mathcal{C}^-$ be a set of well-formed conditions, $A_1, A_2 : \mathcal{K} \nrightarrow \mathcal{Q}$ and $V := \bigcup_{\hat{T}_1 <:_{\Theta,\Lambda} \hat{T}_2 \in C} \{a \mid (a, \_) \in \Theta\}$. Let for all $a \in V$, $A_1(a)$ be well defined. Let $A_2 = \mathrm{Solve}(C, A_1)$ and $S = \{(\kappa, \bigwedge Q) \mid (\kappa, Q) \in A_2\}$.

Show that the then-branch ensures the postcondition

So let $c \in C$ be a condition, such that the if-condition holds. We know that $A_2 = \mathrm{Solve}(C, \mathrm{Weaken}(c, A_1))$. We know that c is a wellformed condition and for all $a \in V$, $A_1(a)$ is defined and therfore the precondition of $\mathrm{Weaken}(c, A_1)$ holds. By Theorem

4.4 $A_2(a)$ is defined for all a for which A_1 is defined, namely for all $a \in V$, thus the precondition of $\text{Solve}(C, \text{Weaken}(c, A_1))$ holds and therefore the postcondition of $\text{Solve}(C, \text{Weaken}(c, A_1))$ holds. In particular we know that for every $a \in V$, $A_2(a) \subseteq \text{Weaken}(c, A_1)$. By Theorem 4.4 we also know that for every $a \in V$, $\text{Weaken}(c, A_1)(a) \subseteq A_1(a)$. Thus for every $a \in V$, $A_2(a) \subseteq A_1(a)$.

We have left to show that for every $a \in V$, if the sharpest solution is generated by subsets of $A_2(a)$, than it is also the sharpest solution over all subsets of $A_1(a)$.

By theorem 4.4 we know that for every $a \in V$, $A_2(a)$ contains the smallest subset of $A_1(a)$ such that the properies hold. Thus $A_2(a)$ is its definition the sharpest solution over all subsets of $A_1(a)$.

Show that the else-branch ensures the postcondition

Once the recursion is done, we can assume that for all $c \in C$, c has the form $(\{\nu : Int | q_1\} <:_{\Theta, \Lambda} \{\nu : Int | [k_2]_{S_2}\})$ and the following holds:

$$(\forall i_1 \in \text{value}_\Gamma(\{\nu : Int | r'_1\}). \ldots . \forall i_n \in \text{value}_\Gamma(\{\nu : Int | r'_n\}).$$
$$[\![r_1 \wedge p]\!]_{\{(b_1, i_1), \ldots, (b_n, i_n)\}} \Rightarrow [\![r_2]\!]_{\{(b_1, i_1), \ldots, (b_n, i_n)\}})$$

for

$$r_2 := \bigwedge [S(\kappa_2)]_{S_2}, \ p := \bigwedge \Lambda,$$

$$r_1 := \begin{cases} \bigwedge [S(k_1)]_{S_1} & \text{if } q_1 \text{ has the form } [k_1]_{S_1} \text{ for } k \in \mathcal{K} \text{ and } S_1 \in \mathcal{V} \nrightarrow \mathcal{Q} \\ q_1 & \text{if } q_1 \in \mathcal{Q} \end{cases},$$

$$\Theta' := \{ \ (a, r)$$
$$\quad | \ r \text{ has the form } q \wedge (a, q) \in \Theta \wedge q \in \mathcal{Q}$$
$$\quad \vee \ r \text{ has the form } [[k]_S]_{S_0} \wedge (a, q) \in \Theta$$
$$\quad\quad \wedge \ q \text{ has the form } [k]_{S_0} \wedge k \in \mathcal{K} \wedge S_0 \in \mathcal{V} \nrightarrow \mathcal{Q}\}$$
$$\{(b_1, r'_1), \ldots, (b_n, r'_n)\} = \Theta'.$$

where $k, k_2 \in \mathcal{K}, S : \mathcal{K} \nrightarrow \mathcal{Q}, Q, \Lambda \subseteq \mathcal{Q}, S_1, S_2 : \mathcal{V} \nrightarrow \mathcal{Q}, q_1 \in \mathcal{K} \cup \mathcal{Q}, \Theta$ be a type variable context $, r_1, p, r_2 \in \mathcal{Q}, a \in \mathcal{V}, \Theta' : \mathcal{V} \nrightarrow \mathcal{Q}, r \in \mathcal{Q}, n \in \mathbb{N}, b_i \in \mathcal{V}, r_i \in \mathcal{Q}$ for $i \in \mathbb{N}_0^n$ and $[t]_A$ denotes the substitution for the term t with a substitution A. This is the same as saying $[\hat{T}_1]_S <:_{\Theta', \Lambda} [\hat{T}_2]_S$ if $[\hat{T}_1]_S \in \mathcal{T}$ and $[\hat{T}_2]_S \in \mathcal{T}$. We know by Theorem 4.4 that $A_2(a)$ is defined for all $a \in V$ and thus $[\hat{T}_1]_S \in \mathcal{T} \wedge [\hat{T}_2]_S \in \mathcal{T}$.

We still need to show that it is the sharpest solution. To do so, we will sketch a proof by contradiction. Assume there exists S' and $a \in V$ such that $S'(a)$ is sharper then $S(a)$. If $S(a) = \textit{False}$ then there can not exist a sharper refinement $S'(a)$ and therefore this is a contradiction. If $S(a) \neq \textit{False}$ then it must have been produced by calling Weaken. This is a contradiction, as by Theorem 4.4 we know that Weaken produces the sharpest solution therefore $S'(a) = S(a)$. $\qquad\square$

Theorem 4.4

Let $c \in \mathcal{C}^-$ be a well-formed condition and therefore c has the form

$$\{\nu : Int | x\} <:_{\Theta,\Lambda} \{\nu : Int | [k_2]_{S_2}\}$$

where x has the form $[k_1]_{S_1}$ or r. Let $A_1, A_2 : \mathcal{K} \nrightarrow \mathcal{Q}$. Let for all $a \in V$, $A_1(a)$ be well defined and $A_2 = \text{Weaken}(c, A_1)$. Let

$$S := \{(k, \bigwedge Q) | (k, Q) \in A_2\}$$

$$r_1 := \begin{cases} \bigwedge[S(k_1)]_{S_1} & \text{if } q_1 \text{ has the form } [k_1]_{S_1} \text{ for } k \in \mathcal{K} \text{ and } S_1 \in \mathcal{V} \nrightarrow \mathcal{Q} \\ q_1 & \text{if } q_1 \in \mathcal{Q} \end{cases}.$$

—

Then the following holds

$$(\forall k \neq k_2. A_1(k) = A_2(k))$$
$$\wedge A_2(k_2) \subseteq A_1(k_2)$$
$$\wedge [k_2]_S \in \mathcal{Q} \wedge \{\nu : Int | r_1\} <:_{\Theta,\Lambda} \{\nu : Int | [[k_2]_S]_{S_2}\}$$
$$\wedge \forall A_2' : \mathcal{V} \nrightarrow \mathcal{P}(\mathcal{Q}).(\forall k \neq k_2. A_1(k) = A_2'(k))$$
$$\wedge A_2'(k_2) \subseteq A_1(k_2)$$
$$\wedge \text{Let } S' := \{(k, \bigwedge Q) | (k, Q) \in A_2'\},$$
$$r_1' := \begin{cases} \bigwedge[S(k_1)]_{S_1} & \text{if } q_1 \text{ has the form } [k_1]_{S_1} \text{ for } k \in \mathcal{K} \text{ and } S_1 \in \mathcal{V} \nrightarrow \mathcal{Q} \\ q_1 & \text{if } q_1 \in \mathcal{Q} \end{cases}$$
$$\text{in } [k_2]_{S'} \in \mathcal{Q}$$
$$\wedge \{\nu : Int | r_1'\} <:_{\Theta,\Lambda} \{\nu : Int | [[k_2]_{S'}]_{S_2}\} \Rightarrow A_2' \subseteq A_2$$

The first post-condition states that by updating A_1 to A_2 only the value for k_2 changes. The second condition states that the updated value $A_2(k_2)$ needs to be a subset of the old value $A_1(k_2)$. The third condition states that the resulting substitution S generates a refinement $[k_1]_S$ that makes the subtyping condition valid. The fourth condition states that the value $A_2(k_2)$ is the smallest subset of $A_1(k_2)$ such that the previous conditions hold.

Proof. Let $c \in \mathcal{C}^-$ be a well-formed condition and therefore c has the form $\{\nu : Int | x\} <:_{\Theta,\Lambda} \{\nu : Int | [k_2]_{S_2}\}$ where x has the form $[k_1]_{S_1}$ or r. Let $A_1, A_2 : \mathcal{K} \nrightarrow \mathcal{Q}$.

Let for all $a \in V$, $A_1(a)$ be well defined and $A_2 = \text{Weaken}(c, A_1)$. Let

$$S := \{(k, \bigwedge Q) | (k, Q) \in A_2\}$$

$$r_1 := \begin{cases} \bigwedge [S(k_1)]_{S_1} & \text{if } q_1 \text{ has the form } [k_1]_{S_1} \text{ for } k \in \mathcal{K} \text{ and } S_1 \in \mathcal{V} \nrightarrow \mathcal{Q} \\ q_1 & \text{if } q_1 \in \mathcal{Q} \end{cases}$$

$$p := \bigwedge \{[q]_S \mid q \in \Lambda\},$$

$$\Theta' := \{ \, (a, r)$$
$$\mid r \text{ has the form } q \wedge (a, q) \in \Theta \wedge q \in \mathcal{Q}$$
$$\vee \; r \text{ has the form } [[k]_S]_{S_0} \wedge (a, q) \in \Theta$$
$$\wedge q \text{ has the form } [k]_{S_0} \wedge k \in \mathcal{K} \wedge S_0 \in \mathcal{V} \nrightarrow \mathcal{Q}\}$$

$$\{(b_1, r_1'), \dots, (b_n, r_n')\} = \Theta'$$

$$Q_2 := \{ \, q$$
$$\mid q \in A(k_2) \wedge \text{wellFormed}(q, \{(b_1, \{\nu : Int|r_1'\}), \dots, (b_n, \{\nu : Int|r_n'\})\})$$
$$\wedge (\forall i_1 \in \text{value}_\Gamma(\{\nu : Int|r_1'\}). \dots \forall i_n \in \text{value}_\Gamma(\{\nu : Int|r_n'\}).$$
$$[\![r_1 \wedge p]\!]_{\{(b_1, i_1), \dots, (b_n, i_n)\}} \Rightarrow [\![q]_{S_2}]\!]_{\{(b_1, i_1), \dots, (b_n, i_n)\}})\}$$

where $k, k_1 \in \mathcal{K}, Q, Q_2 \subseteq \mathcal{Q}, S : \mathcal{K} \nrightarrow \mathcal{Q}, r_1 \in \mathcal{Q}, p \in \mathcal{Q}, S_1, S_2 : \mathcal{V} \nrightarrow \mathcal{Q}, \Theta' : \mathcal{V} \nrightarrow \mathcal{T}, a \in \mathcal{V}, T' \in \mathcal{T} \cup \mathcal{T}^?, n \in \mathbb{N}, b_i \in \mathcal{V}, T_i \in \mathcal{T}$ for $i \in \mathbb{N}_0^n$ and $[t]_A$ denotes the substitution for the term t with a substitution A.

Then

$$A_2 = \{(k, Q) \mid (k, Q) \in A \wedge k \neq k_2\} \cup \{(k_2, Q_2)\}$$

and therefore $\forall k \neq k_2.A_1(k) = A_2(k)$.

We know $A_2(k_2) = Q_2$ and $Q_2 \subseteq A_1(k_2)$ and therefore $A_2(k_2) \subseteq A_1(k_2)$.

We know by the definition of S that $S(k_2) = \bigwedge Q_2$. Therefore it is easy to see that $\{\nu : Int|r_1\} <:_{\Theta, \Lambda} \{\nu : Int|[[k_2]_S]_{S_2}\}$ is true because by definiton of Q_2 it is true for all $q \in Q_2$, thus it is true for $\bigwedge Q_2$ as well.

We finish off by proving that the result is the sharpest, meaning that A_2 is the smallest subset such hat the premise holds. So let there exists a A_2' such that $\forall k \neq k_2.A_1(k) = A_2'(k), A_2'(k_2) \subseteq A_1(k_2), [k_2]_{S'} \in \mathcal{Q}, \{\nu : Int|r_1'\} <:_{\Theta, \Lambda} \{\nu : Int|[[k_2]_{S'}]_{S_2}\}$ and $A_2' \subseteq A_2$ for given S' and r_1.

This means for every $q \in A_2'(k_2)$ that $q \in A_1(k_2)$ and by the definition of subtyping, $\text{wellFormed}(q, \{(b_1, \{\nu : Int|r_1'\}), \dots, (b_n, \{\nu : Int|r_n'\})\})$ and

$$\forall i_1 \in \text{value}_\Gamma(\{\nu : Int|r_1'\}). \dots \forall i_n \in \text{value}_\Gamma(\{\nu : Int|r_n'\}).$$
$$[\![r_1 \wedge p]\!]_{\{(b_1, i_1), \dots, (b_n, i_n)\}} \Rightarrow [\![q]_{S_2}]\!]_{\{(b_1, i_1), \dots, (b_n, i_n)\}}.$$

By the definition of A_2, this means that $q \in A_2(k_2)$. Thus A_2 is the smallest subset. $\qquad \square$