Supplementary Material for "Answer Refinement Modification: Refinement Type System for Algebraic Effects and Handlers"

Fuga Kawamata

Contents

1 Typing Rule for Operation Forwarding

The typing rule for handling constructs presented in Section 3.2 of the main paper assumes that a handler covers all the operations performed by the handled expression. In this section, we present another typing rule for handling constructs to allow *operation forwarding*, that is, allow unhandled operations to be forwarded to outer handlers automatically. The idea of the typing rule is simple: we derive it from an implementation of operation forwarding. As mentioned in Section 3.1 of the main paper, operation forwarding can be implemented in a calculus without forwarding by adding to a handler an operation clause $\mathsf{op}(x, k) \mapsto \textbf{let } y = \mathsf{op}\ x \textbf{ in } k\ y$ for each forwarded operation op. Therefore, we can derive the new typing rule from the typing of the added clauses. The following is the thus derived new typing rule for handling constructs which natively supports operation forwarding:

$$
\frac{
\begin{array}{c}
h = \{\mathbf{return}\ x_r \mapsto c_r, (\mathsf{op}_i(x_i, k_i) \mapsto c_i)_i\} \quad \Gamma \vdash c : \Sigma \triangleright T \ / \ (\forall x_r.C_1) \Rightarrow C_2 \\
\Gamma, x_r : T \vdash c_r : C_1 \quad \left(\Gamma, \widetilde{X_i : \widetilde{B}_i}, x_i : T_{1i}, k_i : (y_i : T_{2i}) \to C_{1i} \vdash c_i : C_{2i}\right)_i \\
\left(\Sigma \ni \mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.(x_i : T_{1i}) \to ((y_i : T_{2i}) \to C_{1i}) \to C_{2i}\right)_i \qquad \mathit{Ops}_{\mathrm{fwd}} = \mathit{dom}(\Sigma) \setminus \mathit{dom}(h) \\
\left(\begin{array}{l}
\Sigma \ni \mathsf{op} : \forall \widetilde{X^{\mathsf{op}} : \widetilde{B^{\mathsf{op}}}}.(x^{\mathsf{op}} : T_1^{\mathsf{op}}) \to \\
\quad ((y^{\mathsf{op}} : T_2^{\mathsf{op}}) \to \Sigma' \triangleright T_0^{\mathsf{op}} \ / \ (\forall z^{\mathsf{op}}.C_0^{\mathsf{op}}) \Rightarrow C_1^{\mathsf{op}}) \to \Sigma' \triangleright T_0^{\mathsf{op}} \ / \ (\forall z^{\mathsf{op}}.C_0^{\mathsf{op}}) \Rightarrow C_2^{\mathsf{op}} \\
\Sigma' \ni \mathsf{op} : \forall \widetilde{X^{\mathsf{op}} : \widetilde{B^{\mathsf{op}}}}.(x^{\mathsf{op}} : T_1^{\mathsf{op}}) \to ((y^{\mathsf{op}} : T_2^{\mathsf{op}}) \to C_1^{\mathsf{op}}) \to C_2^{\mathsf{op}} \quad y^{\mathsf{op}} \notin C_0^{\mathsf{op}} \setminus \{z^{\mathsf{op}}\}
\end{array}\right)_{\mathsf{op} \in \mathit{Ops}_{\mathrm{fwd}}}
\end{array}
}{
\Gamma \vdash \mathbf{with}\ h\ \mathbf{handle}\ c : C_2
}
$$

where $\mathit{dom}(\Sigma)$ denotes the set of the operations associated by Σ and $\mathit{dom}(h)$ denotes the set of the operations handled by h, that is, the set $\{(\mathsf{op}_i)_i\}$. The first two lines are the same as (T-Hndl). The third line is also similar to the last premise of (T-Hndl), but here Σ is allowed to contain operations other than those handled by h. $\mathit{Ops}_{\mathrm{fwd}}$ is exactly the set of the unhandled (i.e., forwarded) operations. The last part is the requirement for the forwarded operations, which can be obtained from the typing derivations of $\mathsf{op}(x, k) \mapsto \mathbf{let}\ y = \mathsf{op}\ x\ \mathbf{in}\ k\ y$ as follows. When we simulate the operation forwarding with the explicit clause, the operation call $\mathsf{op}\ x$ in the clause is handled by an immediate outer handler (we denote it by h' in what follows). Therefore, its operation signature is different from Σ; in fact, it corresponds to Σ' in the rule. Also, the answer types of the original operation calls of op (i.e., the answer types of the operation calls of op in the handled computation c) should have Σ' as their operation signatures, because the final answer type corresponds to the type of the handling construct, which is handled by the immediate outer handler h'. Therefore, the types of the forwarded operations in Σ contains Σ' in their answer types. In addition, the types T_0^{op}, T_1^{op}, T_2^{op}, C_0^{op}, C_1^{op}, and C_2^{op} appear multiple times in Σ and Σ', restricting the type schemes of the operations in $\mathit{Ops}_{\mathrm{fwd}}$. This restriction can be understood as follows. First, assume that the original operation call of op in c has the operation signature Σ such that

$$\Sigma \ni \mathsf{op} : T_1^{\mathsf{op}} \to (T_2^{\mathsf{op}} \to \Sigma' \triangleright T_0^{\mathsf{op}} \ / \ C_0^{\mathsf{op}} \Rightarrow C_1^{\mathsf{op}}) \to \Sigma' \triangleright T_{0A}^{\mathsf{op}} \ / \ C_{0A}^{\mathsf{op}} \Rightarrow C_2^{\mathsf{op}}$$

for some T_1^{op}, T_2^{op}, T_0^{op}, C_0^{op}, C_1^{op}, T_{0A}^{op}, C_{0A}^{op}, C_2^{op}, and Σ', under a context Γ. Here we consider only simple types for simplicity, but a similar argument can be made for dependent and refinement types by appropriately naming the variables like in the rule above. Note that its answer types have Σ' as described earlier, and that we do not impose the restriction yet. From the assumption, the clause $\mathbf{let}\ y = \mathsf{op}\ x\ \mathbf{in}\ k\ y$ should be typed under the context $\Gamma, x : T_1^{\mathsf{op}}, k : T_2^{\mathsf{op}} \to \Sigma' \triangleright T_0^{\mathsf{op}} \ / \ C_0^{\mathsf{op}} \Rightarrow C_1^{\mathsf{op}}$. Then, the input type of op in the clause should be the type of x, namely, T_1^{op}, and the output type of op should be the type of the variable y, which turns out to be T_2^{op} from the type of k. Therefore, the operation signature Σ' for $\mathsf{op}\ x$ should contain $\mathsf{op} : T_1^{\mathsf{op}} \to (T_2^{\mathsf{op}} \to C_{1A}^{\mathsf{op}}) \to C_{2A}^{\mathsf{op}}$ for some C_{1A}^{op} and C_{2A}^{op} . Then, according to the typing rules for operation calls and let-expressions, it is required that $C_{1A}^{\mathsf{op}} = C_1^{\mathsf{op}}$, and the type of $\mathbf{let}\ y = \mathsf{op}\ x\ \mathbf{in}\ k\ y$ is $\Sigma' \triangleright T_0^{\mathsf{op}} \ / \ C_0^{\mathsf{op}} \Rightarrow C_{2A}^{\mathsf{op}}$. Finally, since the type of the clause corresponds to the final answer type of the operation op in Σ (which is $\Sigma' \triangleright T_{0A}^{\mathsf{op}} \ / \ C_{0A}^{\mathsf{op}} \Rightarrow C_2^{\mathsf{op}}$ from the assumption), it should satisfy $T_0^{\mathsf{op}} = T_{0A}^{\mathsf{op}}$, $C_0^{\mathsf{op}} = C_{0A}^{\mathsf{op}}$, and $C_{2A}^{\mathsf{op}} = C_2^{\mathsf{op}}$.

2 Detailed explanation of the benchmark

In this section, we present the result of the verification of the benchmark `queue-2-SAT.ml` as an example. The following is the main part of the program of `queue-2-SAT.ml`:

```
let[@annot_MB "int list ->
    (unit -> ({Get_next: s1, Add_to_queue: s2} |> int option / s => s)) ->
    int option"]
  queue initial (body :unit -> int option) =
    match_with body () {
      retc = (fun x -> (fun _ -> x));
      exnc = raise;
      effc = fun (type a) (e: a eff) -> match e with
        | Get_next _ctx -> Some (fun (k: (a, _) continuation) ->
            (fun queue -> match queue with
              | [] -> continue k None []
              | hd::tl -> continue k (Some hd) tl) )
        | Add_to_queue v -> Some (fun (k: (a, _) continuation) ->
```

```
            (fun queue -> continue k () (queue @ [v])) )
    } initial

let main init =
  queue init (fun () ->
    perform (Add_to_queue 42);
    let _ = perform (Get_next 1(*dummy*)) in
    perform (Get_next 2(*dummy*)) )
```

This program uses two operations `Get_next` and `Add_to_queue`, which are used to dequeue and enqueue elements respectively. The function `queue` manages the queue. It receives an initial queue `initial` and the function `body`, handling the operations performed in `body` in the state-passing manner to simulate the behavior of the queue. The first three lines of the program are the underlying simple type annotation, which tells the function `queue` that the argument `body` may perform the operations `Get_next` and `Add_to_queue` and that its control effect is impure. This annotaion is necessary because our implementation does not support effect polymorphism as mentioned in Section 4 of the main paper. The main function `main` first enqueue one element, and then try to dequeue twice (`Get_next` returns `None` when the queue is empty). Note that we added a ghost parameter `_ctx` to `Get_next`, which is used to distinguish its two occurrences. We give `1` to the first occurrence of `Get_next`, and `2` to the second. This ghost parameter is crucial for the precise verification of this program, described later in this section.

We defined the following refinement type as the specification for the main function `main` (here after, we abbreviate the type int list and int option as ilist and iopt respectively):

$$\{z : \text{ilist} \mid z \neq []\} \to \{z : \text{iopt} \mid z \neq \texttt{None}\}$$

That is, if the queue is initially not empty, the last dequeue should return some value.

By running the verification of the program with the specification, our implementation returns "SAT" as shown in Table 1 in the main paper, that is, the function `main` certainly has the type given as the specification. Let us investigate more detail by seeing the inferred type of the function `queue`:

$$\begin{aligned}
&(\mathit{init} : \{z : \text{ilist} \mid z \neq []\}) \\
&\to (\text{unit} \to \Sigma \triangleright \text{iopt} \ / \ (\forall x.(\text{ilist} \to \{z : \text{iopt} \mid \phi_1\})) \Rightarrow (\{z : \text{ilist} \mid \phi_2\} \to \{z : \text{iopt} \mid z \neq \texttt{None}\})) \\
&\to \{z : \text{iopt} \mid z \neq \texttt{None}\}
\end{aligned}$$

$$\begin{aligned}
\text{where } \Sigma \stackrel{\text{def}}{=} \{&\texttt{Add_to_queue} : \text{int} \to (\text{unit} \to \\
&\quad ((q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{41}\})) \to (\{z : \text{ilist} \mid \phi_2\} \to \{z : \text{iopt} \mid z \neq \texttt{None}\}), \\
&\texttt{Get_next} : (\mathit{ctx} : \text{int}) \to ((y : \text{iopt}) \to \\
&\quad ((q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{31} \wedge \phi_{32}\})) \to ((q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{41} \wedge \phi_{42}\})\}
\end{aligned}$$

$$\begin{aligned}
\phi_1 &\stackrel{\text{def}}{=} \text{isSome}(x) \Rightarrow z \neq \texttt{None} & \phi_2 &\stackrel{\text{def}}{=} \mathit{init} \neq [] \Rightarrow z \neq [] \\
\phi_{31} &\stackrel{\text{def}}{=} \text{isCons}(q) \wedge \text{isSome}(y) \Rightarrow z \neq \texttt{None} & \phi_{32} &\stackrel{\text{def}}{=} \text{isSome}(y) \wedge \mathit{ctx} \geq 2 \Rightarrow z \neq \texttt{None} \\
\phi_{41} &\stackrel{\text{def}}{=} \text{isCons}(q) \wedge \text{isCons}(\text{tail}(q)) \Rightarrow z \neq \texttt{None} & \phi_{42} &\stackrel{\text{def}}{=} \text{isCons}(q) \wedge \mathit{ctx} \geq 2 \Rightarrow z \neq \texttt{None}
\end{aligned}$$

where isSome(x) holds if $x = \texttt{Some}\ v$ for some v, isCons(x) holds if $x = v{::}w$ for some v and w, and tail(x) returns the tail of the list x. In the operation signuture, we can find that `Add_to_queue` changes the answer type from $(q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{41}\}$ to $\{z : \text{ilist} \mid \phi_2\} \to \{z : \text{iopt} \mid z \neq \texttt{None}\}$. Therefore, `perform (Add_to_queue 42)` can be given the control effect

$$(\forall \_.((q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{41}\})) \Rightarrow (\{z : \text{ilist} \mid \phi_2\} \to \{z : \text{iopt} \mid z \neq \texttt{None}\}) \ .$$

Similarly, in the operation signuture, `Get_next` changes the answer type from $(q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{31} \wedge \phi_{32}\}$ to $(q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{41} \wedge \phi_{42}\}$. Here, since the refinements of these answer types contain a condition on ctx, their truth depend on whether $\mathit{ctx} = 1$ (< 2) or $\mathit{ctx} = 2$ (≥ 2). This enables assigning different control effects (i.e., different ARM) to each occurrence of `Get_next` depending on the context. Namely, `perform (Get_next 1)` can be given the control effect

$$(\forall y.(q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{31}\}) \Rightarrow (q : \text{ilist}) \to \{z : \text{iopt} \mid \phi_{41}\}$$

since $\mathit{ctx} = 1$, while `perform (Get_next 2)` can be given the control effect

$$(\forall y.\text{ilist} \to \{z : \text{iopt} \mid \text{isSome}(y) \Rightarrow z \neq \texttt{None}\}) \Rightarrow (q : \text{ilist}) \to \{z : \text{iopt} \mid \text{isCons}(q) \Rightarrow z \neq \texttt{None}\}$$

since $ctx = 2$. Now, the control effect of the argument `body` can be obtained from the composition of these three control effects, which results in

$$(\forall x.(\mathrm{ilist} \to \{z : \mathrm{iopt} \mid \phi_1\})) \Rightarrow (\{z : \mathrm{ilist} \mid \phi_2\} \to \{z : \mathrm{iopt} \mid z \neq \texttt{None}\})\ .$$

Then, the handling construct is assigned the final answer type of `body`, i.e., $\{z : \mathrm{ilist} \mid \phi_2\} \to \{z : \mathrm{iopt} \mid z \neq \texttt{None}\}$, and finally applying the non-empty initial queue to the handling construct returns a value of type $\{z : \mathrm{iopt} \mid z \neq \texttt{None}\}$ as expected.

3 Definitions (other than those shown in the main paper) and Assumptions

3.1 Well-formedness of typing contexts, value types, and computation types

$\boxed{\vdash \Gamma}$ $\boxed{\Gamma \vdash T}$ $\boxed{\Gamma \vdash C}$ $\boxed{\Gamma \vdash \Sigma}$ $\boxed{\Gamma \mid T \vdash S}$

$$\frac{}{\vdash \emptyset}(\text{WE-Empty}) \qquad \frac{\vdash \Gamma \quad x \notin \mathit{dom}(\Gamma) \quad \Gamma \vdash T}{\vdash \Gamma, x : T}(\text{WE-Var}) \qquad \frac{\vdash \Gamma \quad X \notin \mathit{dom}(\Gamma)}{\vdash \Gamma, X : \widetilde{B}}(\text{WE-PVar})$$

$$\frac{\Gamma, x : B \vdash \phi}{\Gamma \vdash \{x : B \mid \phi\}}(\text{WT-Rfn}) \qquad \frac{\Gamma, x : T \vdash C}{\Gamma \vdash (x : T) \to C}(\text{WT-Fun})$$

$$\frac{\Gamma \vdash \Sigma \quad \Gamma \vdash T \quad \Gamma \mid T \vdash S}{\Gamma \vdash \Sigma \triangleright T \ / \ S}(\text{WT-Comp}) \qquad \frac{(\Gamma, \widetilde{X_i : \widetilde{B}_i} \vdash F_i)_i}{\Gamma \vdash \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.F_i)_i\}}(\text{WT-Sig})$$

$$\frac{\vdash \Gamma}{\Gamma \mid T \vdash \square}(\text{WT-Pure}) \qquad \frac{\Gamma, x : T \vdash C_1 \quad \Gamma \vdash C_2}{\Gamma \mid T \vdash (\forall x.C_1) \Rightarrow C_2}(\text{WT-ATM})$$

3.2 Assumptions on well-formedness judgments of formulas, well-formedness judgments of predicates, and semantic validity judgements of formulas

Assumption 1.

- If $\Gamma \vdash \phi$, then $\vdash \Gamma$.
- If $\vdash \Gamma$, $z \notin \mathit{dom}(\Gamma)$ and $\mathit{dom}(\Gamma, z : B) \supseteq \mathit{fv}(\phi)$, then $\Gamma, z : B \vdash \phi$.
- If $\vdash \Gamma, x : T, \Gamma'$ and $\Gamma, \Gamma' \vdash A : \widetilde{B}$, then $\Gamma, x : T, \Gamma' \vdash A : \widetilde{B}$.
- If $\vdash \Gamma, x : T, \Gamma'$ and $\Gamma, \Gamma' \vdash \phi$, then $\Gamma, x : T, \Gamma' \vdash \phi$.
- If $\Gamma, \Gamma' \vDash \phi$, then $\Gamma, x : T, \Gamma' \vDash \phi$.
- If $\Gamma \vdash v : T$ and $\Gamma, x : T, \Gamma' \vdash A : \widetilde{B}$, then $\Gamma, \Gamma'[v/x] \vdash A[v/x] : \widetilde{B}$.
- If $\Gamma \vdash v : T$ and $\Gamma, x : T, \Gamma' \vdash \phi$, then $\Gamma, \Gamma'[v/x] \vdash \phi[v/x]$.
- If $\Gamma \vdash v : T$ and $\Gamma, x : T, \Gamma' \vDash \phi$, then $\Gamma, \Gamma'[v/x] \vDash \phi[v/x]$.
- If $\Gamma \vdash A : \widetilde{B}$ and $\Gamma, X : \widetilde{B}, \Gamma' \vdash A' : \widetilde{B'}$, then $\Gamma, \Gamma'[A/X] \vdash A'[A/X] : \widetilde{B'}$.
- If $\Gamma \vdash A : \widetilde{B}$ and $\Gamma, X : \widetilde{B}, \Gamma' \vdash \phi$, then $\Gamma, \Gamma'[A/X] \vdash \phi[A/X]$.
- If $\Gamma \vdash A : \widetilde{B}$ and $\Gamma, X : \widetilde{B}, \Gamma' \vDash \phi$, then $\Gamma, \Gamma'[A/X] \vDash \phi[A/X]$.
- If $\Gamma \vdash T_1 <: T_2$, $\vdash \Gamma, x : T_1, \Gamma'$ and $\Gamma, x : T_2, \Gamma' \vdash A : \widetilde{B}$, then $\Gamma, x : T_1, \Gamma' \vdash A : \widetilde{B}$.
- If $\Gamma \vdash T_1 <: T_2$, $\vdash \Gamma, x : T_1, \Gamma'$ and $\Gamma, x : T_2, \Gamma' \vdash \phi$, then $\Gamma, x : T_1, \Gamma' \vdash \phi$.
- If $\Gamma \vdash T_1 <: T_2$ and $\Gamma, x : T_2, \Gamma' \vDash \phi$, then $\Gamma, x : T_1, \Gamma' \vDash \phi$.
- If $x \notin \mathit{fv}(\Gamma', \phi)$ and $\Gamma, x : T_0, \Gamma' \vdash \phi$, then $\Gamma, \Gamma' \vdash \phi$.
- If $\Gamma, x : (y : T_1) \to C_1, \Gamma' \vdash \phi$, then $x \notin \mathit{fv}(\Gamma', \phi)$.
- If $\vDash \phi$ and $\Gamma, \phi, \Gamma' \vDash \phi'$, then $\Gamma, \Gamma' \vDash \phi'$.

- If $\Gamma \vdash \phi$, then $\Gamma \models \phi \Rightarrow \phi$.
- If $\Gamma \models \phi_1 \Rightarrow \phi_2$ and $\Gamma \models \phi_2 \Rightarrow \phi_3$, then $\Gamma \models \phi_1 \Rightarrow \phi_3$.
- If $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash \phi$, then $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \models \phi \implies \phi[y/x]$.
- If $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash \phi$, then $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \models \phi[y/x] \implies \phi$.

3.3 Assumptions on primitives

Assumption 2.

- $\vdash \mathit{ty}(p)$ for all p.
- If $\mathit{ty}(p) = (x : T) \to C$, then $\zeta(p, v)$ is defined and $\vdash \zeta(p, v) : C[v/x]$ for all v such that $\vdash v : T$.
- If $\mathit{ty}(p) = \{z : \text{bool} \mid \phi\}$, then $p = \mathbf{true}$ or $p = \mathbf{false}$.

4 Proof of Type Safety

4.1 Progress

Lemma 3 (Weakening)**.**

1. *Assume that* $\vdash \Gamma, x : T_0, \Gamma'$.
   - *If* $\Gamma, \Gamma' \vdash T$, *then* $\Gamma, x : T_0, \Gamma' \vdash T$.
   - *If* $\Gamma, \Gamma' \vdash C$, *then* $\Gamma, x : T_0, \Gamma' \vdash C$.
   - *If* $\Gamma, \Gamma' \vdash \Sigma$, *then* $\Gamma, x : T_0, \Gamma' \vdash \Sigma$.
   - *If* $\Gamma, \Gamma' \mid T \vdash S$, *then* $\Gamma, x : T_0, \Gamma' \mid T \vdash S$.
2. *Assume that* $\vdash \Gamma, x : T_0, \Gamma'$.
   - *If* $\Gamma, \Gamma' \vdash v : T$, *then* $\Gamma, x : T_0, \Gamma' \vdash v : T$.
   - *If* $\Gamma, \Gamma' \vdash c : C$, *then* $\Gamma, x : T_0, \Gamma' \vdash c : C$.
3. - *If* $\Gamma, \Gamma' \vdash T_1 <: T_2$, *then* $\Gamma, x : T_0, \Gamma' \vdash T_1 <: T_2$.
   - *If* $\Gamma, \Gamma' \vdash C_1 <: C_2$, *then* $\Gamma, x : T_0, \Gamma' \vdash C_1 <: C_2$.
   - *If* $\Gamma, \Gamma' \vdash \Sigma_1 <: \Sigma_2$, *then* $\Gamma, x : T_0, \Gamma' \vdash \Sigma_1 <: \Sigma_2$.
   - *If* $\Gamma, \Gamma' \mid T \vdash S_1 <: S_2$, *then* $\Gamma, x : T_0, \Gamma' \mid T \vdash S_1 <: S_2$.

Proof. By simultaneous induction on the derivations. The cases for (WT-Rfn), (T-Op) and (S-Rfn) use Assumption 1. □

Lemma 4 (Narrowing)**.**

1. *Assume that* $\Gamma \vdash T_1 <: T_2$ *and* $\vdash \Gamma, x : T_1, \Gamma'$.
   - *If* $\Gamma, x : T_2, \Gamma' \vdash T$, *then* $\Gamma, x : T_1, \Gamma' \vdash T$.
   - *If* $\Gamma, x : T_2, \Gamma' \vdash C$, *then* $\Gamma, x : T_1, \Gamma' \vdash C$.
   - *If* $\Gamma, x : T_2, \Gamma' \vdash \Sigma$, *then* $\Gamma, x : T_1, \Gamma' \vdash \Sigma$.
   - *If* $\Gamma, x : T_2, \Gamma' \mid T \vdash S$, *then* $\Gamma, x : T_1, \Gamma' \mid T \vdash S$.
2. *Assume that* $\Gamma \vdash T_1 <: T_2$ *and* $\vdash \Gamma, x : T_1, \Gamma'$.
   - *If* $\Gamma, x : T_2, \Gamma' \vdash v : T$, *then* $\Gamma, x : T_1, \Gamma' \vdash v : T$.
   - *If* $\Gamma, x : T_2, \Gamma' \vdash c : C$, *then* $\Gamma, x : T_1, \Gamma' \vdash c : C$.
3. *Assume that* $\Gamma \vdash T_1 <: T_2$.
   - *If* $\Gamma, x : T_2, \Gamma' \vdash T_1' <: T_2'$, *then* $\Gamma, x : T_1, \Gamma' \vdash T_1' <: T_2'$.
   - *If* $\Gamma, x : T_2, \Gamma' \vdash C_1 <: C_2$, *then* $\Gamma, x : T_1, \Gamma' \vdash C_1 <: C_2$.
   - *If* $\Gamma, x : T_2, \Gamma' \vdash \Sigma_1 <: \Sigma_2$, *then* $\Gamma, x : T_1, \Gamma' \vdash \Sigma_1 <: \Sigma_2$.

- *If* $\Gamma, x : T_2, \Gamma' \mid T \vdash S_1 <: S_2$*, then* $\Gamma, x : T_1, \Gamma' \mid T \vdash S_1 <: S_2$.

4. *If* $\Gamma \vdash T_1 <: T_2$ *and* $\Gamma \mid T_2 \vdash S_1 <: S_2$*, then* $\Gamma \mid T_1 \vdash S_1 <: S_2$.

Proof. By simultaneous induction on the derivations. The cases for (WT-Rfn), (T-Op) and (S-Rfn) use Assumption 1. □

Lemma 5 (Substitution)**.**

1. *Assume that* $\Gamma \vdash v : T_0$.
   - *If* $\vdash \Gamma, x : T_0, \Gamma'$*, then* $\vdash \Gamma, \Gamma'[v/x]$.
   - *If* $\Gamma, x : T_0, \Gamma' \vdash T$*, then* $\Gamma, \Gamma'[v/x] \vdash T[v/x]$.
   - *If* $\Gamma, x : T_0, \Gamma' \vdash C$*, then* $\Gamma, \Gamma'[v/x] \vdash C[v/x]$.
   - *If* $\Gamma, x : T_0, \Gamma' \vdash \Sigma$*, then* $\Gamma, \Gamma'[v/x] \vdash \Sigma[v/x]$.
   - *If* $\Gamma, x : T_0, \Gamma' \mid T \vdash S$*, then* $\Gamma, \Gamma'[v/x] \mid T[v/x] \vdash S[v/x]$.
2. *Assume that* $\Gamma \vdash v : T_0$.
   - *If* $\Gamma, x : T_0, \Gamma' \vdash v : T$*, then* $\Gamma, \Gamma'[v/x] \vdash v[v/x] : T[v/x]$.
   - *If* $\Gamma, x : T_0, \Gamma' \vdash c : C$*, then* $\Gamma, \Gamma'[v/x] \vdash c[v/x] : C[v/x]$.
3. *Assume that* $\Gamma \vdash v : T_0$.
   - *If* $\Gamma, x : T_0, \Gamma' \vdash T_1 <: T_2$*, then* $\Gamma, \Gamma'[v/x] \vdash T_1[v/x] <: T_2[v/x]$.
   - *If* $\Gamma, x : T_0, \Gamma' \vdash C_1 <: C_2$*, then* $\Gamma, \Gamma'[v/x] \vdash C_1[v/x] <: C_2[v/x]$.
   - *If* $\Gamma, x : T_0, \Gamma' \vdash \Sigma_1 <: \Sigma_2$*, then* $\Gamma, \Gamma'[v/x] \vdash \Sigma_1[v/x] <: \Sigma_2[v/x]$.
   - *If* $\Gamma, x : T_0, \Gamma' \mid T \vdash S_1 <: S_2$*, then* $\Gamma, \Gamma'[v/x] \mid T \vdash S_1[v/x] <: S_2[v/x]$.

Proof. By simultaneous induction on the derivations. The cases for (WT-Rfn), (T-Op) and (S-Rfn) use Assumption 1. □

Lemma 6 (Predicate Substitution)**.**

1. *Assume that* $\Gamma \vdash A : \widetilde{B}$.
   - *If* $\vdash \Gamma, X : \widetilde{B}, \Gamma'$*, then* $\vdash \Gamma, \Gamma'[A/X]$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash T$*, then* $\Gamma, \Gamma'[A/X] \vdash T[A/X]$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash C$*, then* $\Gamma, \Gamma'[A/X] \vdash C[A/X]$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash \Sigma$*, then* $\Gamma, \Gamma'[A/X] \vdash \Sigma[A/X]$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \mid T \vdash S$*, then* $\Gamma, \Gamma'[A/X] \mid T[A/X] \vdash S[A/X]$.
2. *Assume that* $\Gamma \vdash A : \widetilde{B}$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash v : T$*, then* $\Gamma, \Gamma'[A/X] \vdash v[A/X] : T[A/X]$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash c : C$*, then* $\Gamma, \Gamma'[A/X] \vdash c[A/X] : C[A/X]$.
3. *Assume that* $\Gamma \vdash A : \widetilde{B}$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash T_1 <: T_2$*, then* $\Gamma, \Gamma'[A/X] \vdash T_1[A/X] <: T_2[A/X]$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash C_1 <: C_2$*, then* $\Gamma, \Gamma'[A/X] \vdash C_1[A/X] <: C_2[A/X]$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash \Sigma_1 <: \Sigma_2$*, then* $\Gamma, \Gamma'[A/X] \vdash \Sigma_1[A/X] <: \Sigma_2[A/X]$.
   - *If* $\Gamma, X : \widetilde{B}, \Gamma' \mid T \vdash S_1 <: S_2$*, then* $\Gamma, \Gamma'[A/X] \mid T \vdash S_1[A/X] <: S_2[A/X]$.

Proof. By simultaneous induction on the derivations. The cases for (WT-Rfn), (T-Op) and (S-Rfn) use Assumption 1. □

Lemma 7 (Remove unused type bindings)**.**

- *If* $x \notin fv(\Gamma')$ *and* $\vdash \Gamma, x : T_0, \Gamma'$*, then* $\vdash \Gamma, \Gamma'$.
- *If* $x \notin fv(\Gamma', T)$ *and* $\Gamma, x : T_0, \Gamma' \vdash T$*, then* $\Gamma, \Gamma' \vdash T$.

- *If* $x \notin fv(\Gamma', C)$ *and* $\Gamma, x : T_0, \Gamma' \vdash C$*, then* $\Gamma, \Gamma' \vdash C$*.*
- *If* $x \notin fv(\Gamma', \Sigma)$ *and* $\Gamma, x : T_0, \Gamma' \vdash \Sigma$*, then* $\Gamma, \Gamma' \vdash \Sigma$*.*
- *If* $x \notin fv(\Gamma', T, S)$ *and* $\Gamma, x : T_0, \Gamma' \mid T \vdash S$*, then* $\Gamma, \Gamma' \mid T \vdash S$*.*

Proof. By simultaneous induction on the derivations. The case for (WT-RFN) uses Assumption 1. □

Lemma 8 (Variables of non-refinement types do not occur in types)**.**

- *If* $\vdash \Gamma, x : (y : T_1) \to C_1, \Gamma'$*, then* $x \notin fv(\Gamma')$*.*
- *If* $\Gamma, x : (y : T_1) \to C_1, \Gamma' \vdash T$*, then* $x \notin fv(\Gamma', T)$*.*
- *If* $\Gamma, x : (y : T_1) \to C_1, \Gamma' \vdash C$*, then* $x \notin fv(\Gamma', C)$*.*
- *If* $\Gamma, x : (y : T_1) \to C_1, \Gamma' \vdash \Sigma$*, then* $x \notin fv(\Gamma', \Sigma)$*.*
- *If* $\Gamma, x : (y : T_1) \to C_1, \Gamma' \mid T \vdash S$*, then* $x \notin fv(\Gamma', T, S)$*.*

Proof. By simultaneous induction on the derivations. The case for (WT-RFN) uses Assumption 1. □

Lemma 9 (Remove non-refinement type bindings)**.**

- *If* $\vdash \Gamma, x : (y : T_1) \to C_1, \Gamma'$*, then* $\vdash \Gamma, \Gamma'$*.*
- *If* $\Gamma, x : (y : T_1) \to C_1, \Gamma' \vdash T$*, then* $\Gamma, \Gamma' \vdash T$*.*
- *If* $\Gamma, x : (y : T_1) \to C_1, \Gamma' \vdash C$*, then* $\Gamma, \Gamma' \vdash C$*.*
- *If* $\Gamma, x : (y : T_1) \to C_1, \Gamma' \vdash \Sigma$*, then* $\Gamma, \Gamma' \vdash \Sigma$*.*
- *If* $\Gamma, x : (y : T_1) \to C_1, \Gamma' \mid T \vdash S$*, then* $\Gamma, \Gamma' \mid T \vdash S$*.*

Proof. Immediate by Lemma 8 and 7. □

Lemma 10 (Well-formedness of typing contexts from other judgements)**.**

1. *If* $\Gamma \vdash T$*, then* $\vdash \Gamma$*.*
2. *If* $\Gamma \vdash C$*, then* $\vdash \Gamma$*.*
3. *If* $\Gamma \vdash \Sigma$*, then* $\vdash \Gamma$*.*
4. *If* $\Gamma \mid T \vdash S$*, then* $\vdash \Gamma$*.*

Proof. By simultaneous induction on the derivations. □

Lemma 11 (Well-formedness of types from other judgements)**.**

1. *If* $\Gamma \vdash v : T$*, then* $\Gamma \vdash T$*.*
2. *If* $\Gamma \vdash c : C$*, then* $\Gamma \vdash C$*.*

Proof. By simultaneous induction on the derivations.

1. **Case** (T-CVAR)**:** We have

   (i) $v = x$,
   (ii) $T = \{z : B \mid z = x\}$,
   (iii) $\vdash \Gamma$, and
   (iv) $\Gamma(x) = \{z : B \mid \phi\}$

   for some z, x, and B. W.l.o.g., we can assume that $z \notin dom(\Gamma)$. Also, since (iv) implies $x \in dom(\Gamma)$, it holds that $dom(\Gamma, x : B) \supseteq fv(z = x)$. Then, by the Assumption 1, we have $\Gamma, x : B \vdash z = x$. By (WT-RFN), we have the conclusion.

   Case (T-VAR)**:** We have

   (i) $v = x$,
   (ii) $T = \Gamma(x)$,
   (iii) $\vdash \Gamma$, and
   (iv) $\Gamma(x) \neq \{z : B \mid \phi\}$ for all z, B, and ϕ

for some x. (ii) implies that Γ is of the form $\Gamma_1, x : T, \Gamma_2$ for some Γ_1 and Γ_2. Therefore, by inverting (iii) repeatedly, we have $\Gamma_1 \vdash T$. By Lemma 3 with (iii), we have the conclusion.

Case (T-Prim)**:** We have

(i) $v = p$,

(ii) $T = ty(p)$, and

(iii) $\vdash \Gamma$

for some p. By Assumption 2, we have $\vdash ty(p)$. By Lemma 3 with (iii), we have the conclusion.

Case (T-Fun)**:** We have

(i) $v = \mathbf{rec}(f, x).c$,

(ii) $T = (x : T_0) \to C$, and

(iii) $\Gamma, x : T_0 \vdash c : C$

for some f, x, c, T_0, and C. By the IH of (iii), we have $\Gamma, f : (x : T_0) \to C, x : T_0 \vdash C$. By Lemma 9, we have $\Gamma, x : T_0 \vdash C$. By (WT-Fun), we have the conclusion.

Case (T-VSub)**:** Immediate by inversion.

2. **Case** (T-Ret)**:** We have

(i) $c = \mathbf{return}\ v$,

(ii) $C = \emptyset \triangleright T \ / \ \square$, and

(iii) $\Gamma \vdash v : T$

for some v and T. By the IH of (iii), we have $\Gamma \vdash T$. By Lemma 10, we have $\vdash \Gamma$. Then, we have the conclusion by the following derivation:

$$\dfrac{\dfrac{}{\Gamma \vdash \emptyset} \quad \Gamma \vdash T \quad \dfrac{\vdash \Gamma}{\Gamma \mid T \vdash \square}}{\Gamma \vdash \emptyset \triangleright T \ / \ \square}$$

Case (T-App)**:** We have

(i) $c = v_1\ v_2$,

(ii) $C = C_0[v_2/x]$,

(iii) $\Gamma \vdash v_1 : (x : T_0) \to C_0$, and

(iv) $\Gamma \vdash v_2 : T_0$

for some x, v_1, v_2, T_0 and C_0. By the IH of (iii), we have $\Gamma \vdash (x : T_0) \to C_0$. By inversion, we have $\Gamma, x : T_0 \vdash C_0$. By Lemma 5, we have the conclusion.

Case (T-If)**:** We have

(i) $c = \mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2$,

(ii) $\Gamma \vdash v : \{x : \mathrm{bool} \mid \phi\}$,

(iii) $\Gamma, v = \mathbf{true} \vdash c_1 : C$, and

(iv) $\Gamma, v = \mathbf{false} \vdash c_2 : C$

for some x, v, c_1, c_2, and ϕ. By the IH of (iii), we have $\Gamma, v = \mathbf{true} \vdash C$. By Lemma 7, we have the conclusion.

Case (T-CSub)**:** Immediate by inversion.

Case (T-LetP)**:** We have

(i) $c = \mathbf{let}\ x = c_1\ \mathbf{in}\ c_2$,

(ii) $C = \Sigma \triangleright T_2 \ / \ \square$,

(iii) $\Gamma \vdash c_1 : \Sigma \triangleright T_1 \ / \ \square$,

(iv) $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T_2 \ / \ \square$, and

(v) $x \notin fv(T_2) \cup fv(\Sigma)$

for some x, c_1, c_2, Σ, T_1, and T_2. By the IHs of (iii) and (iv) respectively, we have

- $\Gamma \vdash \Sigma \triangleright T_1 \ / \ \square$ and
- $\Gamma, x : T_1 \vdash \Sigma \triangleright T_2 \ / \ \square$.

By inversion, we have

(vi) $\Gamma \vdash \Sigma$, and

(vii) $\Gamma, x : T_1 \vdash T_2$.

By Lemma 7 with (v) (vii), we have

(viii) $\Gamma \vdash T_2$.

By Lemma 10 with (vi), we have $\vdash \Gamma$. From this fact and (vi) and (viii), we have the conclusion by the following derivation:

$$\dfrac{\Gamma \vdash \Sigma \quad \Gamma \vdash T_2 \quad \dfrac{\vdash \Gamma}{\Gamma \mid T_2 \vdash \square}}{\Gamma \vdash \Sigma \triangleright T_2 \ / \ \square}$$

Case (T-LetIp)**:** We have

(i) $c = \mathbf{let}\ x = c_1\ \mathbf{in}\ c_2$,

(ii) $C = \Sigma \triangleright T_2 \ / \ (\forall z.C_{21}) \Rightarrow C_{12}$,

(iii) $\Gamma \vdash c_1 : \Sigma \triangleright T_1 \ / \ (\forall x.C_0) \Rightarrow C_{12}$,

(iv) $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T_2 \ / \ (\forall z.C_{21}) \Rightarrow C_0$, and

(v) $x \notin \mathit{fv}(T_2) \cup \mathit{fv}(\Sigma) \cup (\mathit{fv}(C_{21}) \setminus \{z\})$

for some $x, c_1, c_2, \Sigma, T_1, T_2, C_0, C_{12}$ and C_{21}. By the IHs of (iii) and (iv) respectively, we have

- $\Gamma \vdash \Sigma \triangleright T_1 \ / \ (\forall x.C_0) \Rightarrow C_{12}$ and
- $\Gamma, x : T_1 \vdash \Sigma \triangleright T_2 \ / \ (\forall z.C_{21}) \Rightarrow C_0$.

By inversion, we have

(vi) $\Gamma \vdash \Sigma$,

(vii) $\Gamma \mid T_1 \vdash (\forall x.C_0) \Rightarrow C_{12}$,

(viii) $\Gamma, x : T_1 \vdash T_2$, and

(ix) $\Gamma, x : T_1 \mid T_2 \vdash (\forall z.C_{21}) \Rightarrow C_0$.

By Lemma 7 with (v) (viii), we have

(x) $\Gamma \vdash T_2$.

By inversion with (vii) and (ix) respectively, we have

(xi) $\Gamma, x : T_1 \vdash C_0$,

(xii) $\Gamma \vdash C_{12}$, and

(xiii) $\Gamma, x : T_1, z : T_2 \vdash C_{21}$.

W.l.o.g., we can assume $x \neq z$. Then, (v) implies $x \notin \mathit{fv}(C_{21})$. Therefore, by Lemma 7 with (xiii) and (v), we have $\Gamma, z : T_2 \vdash C_{21}$. From this and (vi), (x), and (xii), we have the conclusion by the following derivation:

$$\dfrac{\Gamma \vdash \Sigma \quad \Gamma \vdash T_2 \quad \dfrac{\Gamma, z : T_2 \vdash C_{21} \quad \Gamma \vdash C_{12}}{\Gamma \mid T_2 \vdash (\forall z.C_{21}) \Rightarrow C_{12}}}{\Gamma \vdash \Sigma \triangleright T_2 \ / \ (\forall z.C_{21}) \Rightarrow C_{12}}$$

Case (T-Op)**:** We have

(i) $c = \mathsf{op}\ v$,

(ii) $C = \Sigma \triangleright T_2[\widetilde{A/X}][v/x] \ / \ (\forall y.C_1[\widetilde{A/X}][v/x]) \Rightarrow C_2[\widetilde{A/X}][v/x]$,

(iii) $\Sigma \ni \mathsf{op} : \forall \widetilde{X : \widetilde{B}}.(x : T_1) \to ((y : T_2) \to C_1) \to C_2$,

(iv) $\Gamma \vdash \Sigma$,

(v) $\Gamma \vdash \widetilde{A : \widetilde{B}}$, and

(vi) $\Gamma \vdash v : T_1[\widetilde{A/X}]$

for some $x, y, v, \widetilde{X}, \widetilde{A}, \widetilde{\widetilde{B}}, \Sigma, T_1, T_2, C_1$ and C_2. By inversion of (iv) with (iii), we have

$$\Gamma, \widetilde{X : \widetilde{B}} \vdash (x : T_1) \to ((y : T_2) \to C_1) \to C_2 \ .$$

By more inversion and Lemma 9, we have

- $\Gamma, \widetilde{X : \widetilde{B}}, x : T_1 \vdash T_2$,
- $\Gamma, \widetilde{X : \widetilde{B}}, x : T_1, y : T_2 \vdash C_1$, and
- $\Gamma, \widetilde{X : \widetilde{B}}, x : T_1 \vdash C_2$.

By Lemma 6 with (v) and Lemma 5 with (vi), we have

- $\Gamma \vdash T_2[\widetilde{A/X}][v/x]$,
- $\Gamma, y : T_2[\widetilde{A/X}][v/x] \vdash C_1[\widetilde{A/X}][v/x]$, and
- $\Gamma \vdash C_2[\widetilde{A/X}][v/x]$.

From these and (iv), we have the conclusion by the following derivation:

$$\dfrac{\Gamma \vdash \Sigma \quad \Gamma \vdash T_2[\widetilde{A/X}][v/x] \quad \dfrac{\Gamma, y : T_2[\widetilde{A/X}][v/x] \vdash C_1[\widetilde{A/X}][v/x] \quad \Gamma \vdash C_2[\widetilde{A/X}][v/x]}{\Gamma \mid T_2[\widetilde{A/X}][v/x] \vdash (\forall y.C_1[\widetilde{A/X}][v/x]) \Rightarrow C_2[\widetilde{A/X}][v/x]}}{\Gamma \vdash \Sigma \triangleright T_2[\widetilde{A/X}][v/x] \;/\; (\forall y.C_1[\widetilde{A/X}][v/x]) \Rightarrow C_2[\widetilde{A/X}][v/x]}$$

Case (T-Hndl)**:** We have

(i) $c = \textbf{with}\ h\ \textbf{handle}\ c_0$,
(ii) $\Gamma \vdash c_0 : \Sigma \triangleright T \;/\; (\forall x_r.C_1) \Rightarrow C$

for some x_r, h, c_0, Σ, T and C_1. We have the conclusion by applying inversion twice to (ii).

□

Lemma 12 (Canonical forms)**.**

1. *If* $\vdash v : (x : T) \to C$*, then (i)* $v = \textbf{rec}(f, x).c$ *for some* f, c*, or (ii)* $v = p$ *for some* p *and* $\zeta(p, v)$ *is defined for all* v *such that* $\vdash v : T$*.*
2. *If* $\vdash v : \{x : \mathrm{bool} \mid \phi\}$*, then* $v = \textbf{true}$ *or* $v = \textbf{false}$*.*

Proof. By induction on the derivations.

1. **Case** (T-Fun)**:** Obvious.
   Case (T-Prim)**:** Immediate from Assumption 2.
   Case (T-VSub)**:** By the IH and inversion of the subtyping judgment. The case for (ii) uses Lemma 11.
   Otherwise: Contradictory.
2. **Case** (T-Prim)**:** Immediate from Assumption 2.
   Case (T-VSub)**:** By the IH and inversion of the subtyping judgment.
   Otherwise: Contradictory.

□

Theorem 13 (Progress)**.** *If* $\emptyset \vdash c : \Sigma \triangleright T \;/\; S$*, then either*

- $c = \textbf{return}\ v$ *for some* v *such that* $\emptyset \vdash v : T$*,*
- $c = K[\mathsf{op}\ v]$ *for some* K, op *and* v *such that* $\mathsf{op} \in dom(\Sigma)$*, or*
- $c \longrightarrow c'$ *for some* c'*.*

Proof. By induction on the derivation.

Case (T-Ret) **and** (T-Op)**:** Obvious.

Case (T-CSub)**:** By the IH. Note that $\vdash \Sigma' <: \Sigma$ implies $dom(\Sigma') \supseteq dom(\Sigma)$.

Case (T-App)**:** We have

(i) $c = v_1\ v_2$,
(ii) $\vdash v_1 : (x : T_1) \to \Sigma \triangleright T \;/\; S$, and
(iii) $\vdash v_2 : T_1$

for some v_1, v_2, x, and T_1. By Lemma 12 with (ii), either one of the following two cases holds.

- $v_1 = \textbf{rec}(f, x).c_1$ for some f, c_1:
  By (E-App), we have $(\textbf{rec}(f, x).c_1)\ v_2 \longrightarrow c_1[v_2/x][(\textbf{rec}(f, x).c_1)/f]$.

- $v_1 = p$ for some p and $\zeta(p, v)$ is defined for all v such that $\vdash v : T_1$:
  As (iii) holds, $\zeta(p, v_2)$ is defined. Therefore, by (E-Prim) we have $p\ v_2 \longrightarrow \zeta(p, v_2)$.

Case (T-If)**:** We have

(i) $c = \textbf{if}\ v\ \textbf{then}\ c_1\ \textbf{else}\ c_2$,
(ii) $\vdash v : \{x : \text{bool} \mid \phi\}$,
(iii) $v = \textbf{true} \vdash c_1 : \Sigma \triangleright T \ /\ S$, and
(iv) $v = \textbf{false} \vdash c_2 : \Sigma \triangleright T \ /\ S$

for some v, c_1, c_2, x, and ϕ. By Lemma 12 with (ii), either one of the following two cases holds.

- $v = \textbf{true}$: By (E-IfT), we have $\textbf{if true then}\ c_1\ \textbf{else}\ c_2 \longrightarrow c_1$.
- $v = \textbf{false}$: By (E-IfF), we have $\textbf{if false then}\ c_1\ \textbf{else}\ c_2 \longrightarrow c_2$.

Case (T-LetP)**:** We have

(i) $c = \textbf{let}\ x = c_1\ \textbf{in}\ c_2$, and
(ii) $\vdash c_1 : \Sigma \triangleright T_1 \ /\ \square$

for some x, c_1, c_2, and T_1. By the IH of (ii), either one of the following three cases holds.

- $c_1 = \textbf{return}\ v_1$ for some v_1:
  By (E-LetRet), we have $\textbf{let}\ x = \textbf{return}\ v_1\ \textbf{in}\ c_2 \longrightarrow c_2[v_1/x]$.
- $c_1 = K_1[\mathsf{op}\ v_1]$ for some K_1, op and v_1 s.t. $\mathsf{op} \in \mathit{dom}(\Sigma)$:
  We have the conclusion with $c = K[\mathsf{op}\ v_1]$ where $K = \textbf{let}\ x = K_1\ \textbf{in}\ c_1$.
- $c_1 \longrightarrow c_1'$ for some c_1':
  By (E-Let), we have $\textbf{let}\ x = c_1\ \textbf{in}\ c_2 \longrightarrow \textbf{let}\ x = c_1'\ \textbf{in}\ c_2$.

Case (T-LetIp)**:** Similar to the case for (T-LetP).

Case (T-Hndl)**:** We have

(i) $c = \textbf{with}\ h\ \textbf{handle}\ c_0$,
(ii) $h = \{\textbf{return}\ x_r \mapsto c_r, (\mathsf{op}_i(x_i, k_i) \mapsto c_i)_i\}$,
(iii) $\Sigma_0 = \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.(x_i : T_{1i}) \to ((y_i : T_{2i}) \to C_{1i}) \to C_{2i})_i\}$, and
(iv) $\vdash c_0 : \Sigma_0 \triangleright T_0 \ /\ (\forall x_r.C_1) \Rightarrow (\Sigma \triangleright T \ /\ S)$

for some $c_0, x_r, c_r, (\mathsf{op}_i)_i, (x_i)_i, (k_i)_i, (c_i)_i, (\widetilde{X_i})_i, (\widetilde{\widetilde{B}_i})_i, (T_{1i})_i, (T_{2i})_i, (C_{1i})_i, (C_{2i})_i, \Sigma_0, T_0$, and C_1. By the IH of (iv), either one of the following three cases holds.

- $c_0 = \textbf{return}\ v_0$ for some v_0:
  By (E-HndlRet), we have $\textbf{with}\ h\ \textbf{handle return}\ v_0 \longrightarrow c_r[v_0/x_r]$.
- $c_0 = K_0[\mathsf{op}\ v_0]$ for some K_0, op, and v_0 s.t. $\mathsf{op} \in \mathit{dom}(\Sigma_0)$:
  Since $\mathsf{op} \in \mathit{dom}(\Sigma_0) = \{(\mathsf{op}_i)_i\}$, there exists some j such that $1 \le j \le |\mathit{dom}(\Sigma)|$ and $\mathsf{op} = \mathsf{op}_j$. Then, by (E-HndlOp) we have

$$\textbf{with}\ h\ \textbf{handle}\ K_0[\mathsf{op}_j\ v_0] \longrightarrow c_j[v_0/x_j][\lambda y.\textbf{with}\ h\ \textbf{handle}\ K_0[\textbf{return}\ y]/k_j]\ .$$

- $c_0 \longrightarrow c_0'$ for some c_0':
  By (E-Hndl), we have $\textbf{with}\ h\ \textbf{handle}\ c_0 \longrightarrow \textbf{with}\ h\ \textbf{handle}\ c_0'$.

$\square$

4.2 Subject Reduction

Lemma 14 (Remove tautology)**.**

1. *If* $\vdash \Gamma, \phi, \Gamma'$, *then* $\vdash \Gamma, \Gamma'$.
2. - *If* $\Gamma, \phi, \Gamma' \vdash T$, *then* $\Gamma, \Gamma' \vdash T$.
   - *If* $\Gamma, \phi, \Gamma' \vdash C$, *then* $\Gamma, \Gamma' \vdash C$.
   - *If* $\Gamma, \phi, \Gamma' \vdash \Sigma$, *then* $\Gamma, \Gamma' \vdash \Sigma$.
   - *If* $\Gamma, \phi, \Gamma' \mid T \vdash S$, *then* $\Gamma, \Gamma' \mid T \vdash S$.
3. *Assume that* $\models \phi$.
   - *If* $\Gamma, \phi, \Gamma' \vdash v : T$, *then* $\Gamma, \Gamma' \vdash v : T$.
   - *If* $\Gamma, \phi, \Gamma' \vdash c : C$, *then* $\Gamma, \Gamma' \vdash c : C$.
4. *Assume that* $\models \phi$.
   - *If* $\Gamma, \phi, \Gamma' \vdash T'_1 <: T'_2$, *then* $\Gamma, \Gamma' \vdash T'_1 <: T'_2$.
   - *If* $\Gamma, \phi, \Gamma' \vdash C_1 <: C_2$, *then* $\Gamma, \Gamma' \vdash C_1 <: C_2$.
   - *If* $\Gamma, \phi, \Gamma' \vdash \Sigma_1 <: \Sigma_2$, *then* $\Gamma, \Gamma' \vdash \Sigma_1 <: \Sigma_2$.
   - *If* $\Gamma, \phi, \Gamma' \mid T \vdash S_1 <: S_2$, *then* $\Gamma, \Gamma' \mid T \vdash S_1 <: S_2$.

Proof.

1. Immediate by Lemma 7.
2. Immediate by Lemma 7.
3. By simultaneous induction on the derivations.
4. By simultaneous induction on the derivations. The case for (S-Rfn) uses Assumption 1.

□

Lemma 15 (Reflexivity of subtyping)**.**

1. *If* $\Gamma \vdash T$, *then* $\Gamma \vdash T <: T$.
2. *If* $\Gamma \vdash C$, *then* $\Gamma \vdash C <: C$.
3. *If* $\Gamma \vdash \Sigma$, *then* $\Gamma \vdash \Sigma <: \Sigma$.
4. *If* $\Gamma \mid T \vdash S$, *then* $\Gamma \mid T \vdash S <: S$.

Proof. By simultaneous induction on the derivations. The case for (WT-Rfn) uses Assumption 1. □

Lemma 16 (Transitivity of subtyping)**.**

1. *If* $\Gamma \vdash T_1 <: T_2$ *and* $\Gamma \vdash T_2 <: T_3$, *then* $\Gamma \vdash T_1 <: T_3$.
2. *If* $\Gamma \vdash C_1 <: C_2$ *and* $\Gamma \vdash C_2 <: C_3$, *then* $\Gamma \vdash C_1 <: C_3$.
3. *If* $\Gamma \vdash \Sigma_1 <: \Sigma_2$ *and* $\Gamma \vdash \Sigma_2 <: \Sigma_3$, *then* $\Gamma \vdash \Sigma_1 <: \Sigma_3$.
4. *If* $\Gamma \mid T \vdash S_1 <: S_2$ *and* $\Gamma \mid T \vdash S_2 <: S_3$, *then* $\Gamma \mid T \vdash S_1 <: S_3$.

Proof. By simultaneous induction on the structure of T_2, C_2, Σ_2 and S_2.

1. Case analysis on $\Gamma \vdash T_1 <: T_2$.

   Case (S-Rfn)**:** By inversion, Assumption 1 and (S-Rfn).

   Case (S-Fun)**:** By inversion, the IHs, Lemma 4, and (S-Fun).

2. By inversion of the both derivations, we have

   (i) $\Sigma_1 = \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.F_{1i})_i, (\mathsf{op}'_i : \forall \widetilde{X'_i : \widetilde{B'}_i}.F'_{1i})_i, (\mathsf{op}''_i : \forall \widetilde{X''_i : \widetilde{B''}_i}.F''_{1i})_i\}$,

   (ii) $\Sigma_2 = \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.F_{2i})_i, (\mathsf{op}'_i : \forall \widetilde{X'_i : \widetilde{B'}_i}.F'_{2i})_i\}$,

(iii) $\Sigma_3 = \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.F_{2i})_i\}$,

(iv) $(\Gamma, \widetilde{X_i : \widetilde{B}_i} \vdash F_{1i} <: F_{2i})_i$,

(v) $(\Gamma, \widetilde{X_i : \widetilde{B}_i} \vdash F_{2i} <: F_{3i})_i$, and

(vi) $(\Gamma, \widetilde{X'_i : \widetilde{B'}_i} \vdash F'_{1i} <: F'_{2i})_i$.

By the IH with (iv) and (v), we have $(\Gamma, \widetilde{X_i : \widetilde{B}_i} \vdash F_{1i} <: F_{3i})_i$. By (S-Sig), we have the conclusion.

3. By inversion, the IHs, Lemma 4, and (S-Comp).

4. Case analysis on $\Gamma \vdash S_1 <: S_2$.

Case (S-Pure)**:** Since we have $S_1 = \square = S_2$, we have the conclusion immediately from $\Gamma \vdash S_2 <: S_3$.

Case (S-ATM)**:** We have

(i) $S_1 = (\forall x.C_{11}) \Rightarrow C_{12}$,
(ii) $S_2 = (\forall x.C_{21}) \Rightarrow C_{22}$,
(iii) $\Gamma, x : T \vdash C_{21} <: C_{11}$, and
(iv) $\Gamma \vdash C_{12} <: C_{22}$

for some $x, C_{11}, C_{12}, C_{21}$, and C_{22}. Since (ii), the only rule applicable to $\Gamma \vdash S_2 <: S_3$ is (S-ATM). Therefore, by inversion we have

(v) $S_3 = (\forall x.C_{31}) \Rightarrow C_{32}$,
(vi) $\Gamma, x : T \vdash C_{31} <: C_{21}$, and
(vii) $\Gamma \vdash C_{22} <: C_{32}$

for some C_{31} and C_{32}. By the IHs, we have

- $\Gamma, x : T \vdash C_{31} <: C_{11}$ and
- $\Gamma \vdash C_{12} <: C_{32}$.

We have the conclusion by (S-ATM).

Case (S-Embed)**:** We have

(i) $S_1 = \square$,
(ii) $S_2 = (\forall x.C_{21}) \Rightarrow C_{22}$,
(iii) $\Gamma, x : T \vdash C_{21} <: C_{22}$, and
(iv) $x \notin fv(C_{22})$

for some x, C_{21}, and C_{22}. Since (ii), the only rule applicable to $\Gamma \vdash S_2 <: S_3$ is (S-ATM). Therefore, by inversion we have

(v) $S_3 = (\forall x.C_{31}) \Rightarrow C_{32}$,
(vi) $\Gamma, x : T \vdash C_{31} <: C_{21}$, and
(vii) $\Gamma \vdash C_{22} <: C_{32}$

for some C_{31} and C_{32}. W.l.o.g., we can assume that $x \notin fv(C_{32})$. Then, by Lemma 3 with (vii), we have

(viii) $\Gamma, x : T \vdash C_{22} <: C_{32}$.

By the IHs with (iii), (iii) and (viii), we have $\Gamma, x : T \vdash C_{31} <: C_{32}$. Then we have the conclusion by (S-Embed).

$\square$

Lemma 17 (Subtyping with equal variables)**.**

- *If* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash T$, *then* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash T <: T[y/x]$.
- *If* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash T$, *then* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash T[y/x] <: T$.
- *If* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash C$, *then* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash C <: C[y/x]$.
- *If* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash C$, *then* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash C[y/x] <: C$.
- *If* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash \Sigma$, *then* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash \Sigma <: \Sigma[y/x]$.

- *If* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash \Sigma$, *then* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \vdash \Sigma[y/x] <: \Sigma$.
- *If* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \mid T \vdash S$, *then* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \mid T \vdash S <: S[y/x]$.
- *If* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \mid T \vdash S$, *then* $\Gamma, x : \{z : B \mid z = y\}, \Gamma' \mid T \vdash S[y/x] <: S$.

Proof. By simultaneous induction on the derivations. The case for (WT-RFN) uses Assumption 1. The cases for (WT-FUN) and (WT-COMP) uses Lemma 4. □

Lemma 18 (Inversion)**.**

1. *If* $\Gamma \vdash p : T$, *then*
   - $\Gamma \vdash ty(p) <: T$, *and*
   - $\Gamma \vdash p : ty(p)$.
2. *If* $\Gamma \vdash \mathbf{rec}(f, x).c : (x : T) \to C$, *then there exist some* T_0 *and* C_0 *such that*
   - $\Gamma \vdash \mathbf{rec}(f, x).c : (x : T_0) \to C_0$,
   - $\Gamma \vdash (x : T_0) \to C_0 <: (x : T) \to C$, *and*
   - $\Gamma, f : (x : T_0) \to C_0, x : T_0 \vdash c : C_0$.
3. *If* $\Gamma \vdash \mathbf{return}\ v : \Sigma \triangleright T \ / \ S$, *then there exist some* T' *such that*
   - $\Gamma \vdash T' <: T$,
   - $\Gamma \vdash v : T'$, *and*
   - $\Gamma \mid T' \vdash \square <: S$.
4. *If* $\Gamma \vdash \mathsf{op}\ v : \Sigma \triangleright T \ / \ S$, *then there exist some* $\widetilde{X}, \widetilde{\widetilde{B}}, \widetilde{A}, x, y, T_1, T_2, C_1, C_2, C_{01}$, *and* C_{02} *such that*
   - $S = (\forall y.C_{01}) \Rightarrow C_{02}$,
   - $\Sigma \ni \mathsf{op} : \forall \widetilde{X : \widetilde{B}}.(x : T_1) \to ((y : T_2) \to C_1) \to C_2$,
   - $\Gamma \vdash \widetilde{A : \widetilde{B}}$,
   - $\Gamma \vdash v : T_1[\widetilde{A/X}]$,
   - $\Gamma \vdash T_2[\widetilde{A/X}][v/x] <: T$,
   - $\Gamma, y : T_2[\widetilde{A/X}][v/x] \vdash C_{01} <: C_1[\widetilde{A/X}][v/x]$, *and*
   - $\Gamma \vdash C_2[\widetilde{A/X}][v/x] <: C_{02}$.
5. *If* $\Gamma \vdash \mathbf{let}\ x = c_1\ \mathbf{in}\ c_2 : \Sigma \triangleright T \ / \ \square$, *then there exists some* T_1 *such that*
   - $\Gamma \vdash c_1 : \Sigma \triangleright T_1 \ / \ \square$,
   - $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T \ / \ \square$, *and*
   - $x \notin fv(T) \cup fv(\Sigma)$.
6. *If* $\Gamma \vdash \mathbf{let}\ x = c_1\ \mathbf{in}\ c_2 : \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C_2$, *then there exist some* T_1 *and* C_0 *such that*
   - $\Gamma \vdash c_1 : \Sigma \triangleright T_1 \ / \ (\forall x.C_0) \Rightarrow C_2$,
   - $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C_0$, *and*
   - $x \notin fv(T) \cup fv(\Sigma) \cup (fv(C_1) \setminus \{z\})$.

Proof. By induction on the derivations.

1. Straightforward with Lemma 15 and 16.
2. Straightforward with Lemma 15 and 16.
3. Straightforward with Lemma 15 and 16.
4. **Case** (T-OP)**:** Obvious with Lemma 15.

   Case (T-CSUB)**:** We have

   (i) $\Gamma \vdash \mathsf{op}\ v : \Sigma' \triangleright T' \ / \ S'$,

(ii) $\Gamma \vdash \Sigma' \triangleright T' \;/\; S' <: \Sigma \triangleright T \;/\; S$, and

(iii) $\Gamma \vdash \Sigma \triangleright T \;/\; S$

for some Σ', T', and S'. By the IH on (i), we have

(iv) $S' = (\forall y.C'_{01}) \Rightarrow C'_{02}$,

(v) $\Sigma' \ni \mathsf{op} : \forall \widetilde{X : \widetilde{B}}.(x : T'_1) \to ((y : T'_2) \to C'_1) \to C'_2$,

(vi) $\Gamma \vdash \widetilde{A : \widetilde{B}}$,

(vii) $\Gamma \vdash v : T'_1[\widetilde{A/X}]$,

(viii) $\Gamma \vdash T'_2[\widetilde{A/X}][v/x] <: T'$,

(ix) $\Gamma, y : T'_2[\widetilde{A/X}][v/x] \vdash C'_{01} <: C'_1[\widetilde{A/X}][v/x]$, and

(x) $\Gamma \vdash C'_2[\widetilde{A/X}][v/x] <: C'_{02}$

for some $\widetilde{X}, \widetilde{\widetilde{B}}, \widetilde{A}, x, y, T'_1, T'_2, C'_1, C'_2, C'_{01}$, and C'_{02}. By inversion of (ii), we have

(xi) $\Gamma \vdash \Sigma <: \Sigma'$,

(xii) $\Gamma \vdash T' <: T$, and

(xiii) $\Gamma \mid T' \vdash S' <: S$.

By inversion of (xiii) with (iv), we have

(xiv) $S = (\forall y.C_{01}) \Rightarrow C_{02}$,

(xv) $\Gamma, y : T' \vdash C_{01} <: C'_{01}$, and

(xvi) $\Gamma \vdash C'_{02} <: C_{02}$

for some C_{01} and C_{02}. On the other hand, by inversion of (xi) with (v) and Lemma 9, we have

(xvii) $\Sigma \ni \mathsf{op} : \forall \widetilde{X : \widetilde{B}}.(x : T_1) \to ((y : T_2) \to C_1) \to C_2$,

and

- $\Gamma, \widetilde{X : \widetilde{B}} \vdash T'_1 <: T_1$,
- $\Gamma, \widetilde{X : \widetilde{B}}, x : T'_1 \vdash T_2 <: T'_2$,
- $\Gamma, \widetilde{X : \widetilde{B}}, x : T'_1, y : T_2 \vdash C'_1 <: C_1$, and
- $\Gamma, \widetilde{X : \widetilde{B}}, x : T'_1 \vdash C_2 <: C'_2$

for some T_1, T_2, C_1 and C_2. Then, by Lemma 6 with (vi) and Lemma 5 with (vii), we have

(xviii) $\Gamma \vdash T'_1[\widetilde{A/X}] <: T_1[\widetilde{A/X}]$,

(xix) $\Gamma \vdash T_2[\widetilde{A/X}][v/x] <: T'_2[\widetilde{A/X}][v/x]$,

(xx) $\Gamma, y : T_2[\widetilde{A/X}][v/x] \vdash C'_1[\widetilde{A/X}][v/x] <: C_1[\widetilde{A/X}][v/x]$, and

(xxi) $\Gamma \vdash C_2[\widetilde{A/X}][v/x] <: C'_2[\widetilde{A/X}][v/x]$.

By subsumption of (vii) with (xviii),

(xxii) $\Gamma \vdash v : T_1[\widetilde{A/X}]$.

By Lemma 16 with (xix), (viii) and (xii), we have

(xxiii) $\Gamma \vdash T_2[\widetilde{A/X}][v/x] <: T'$ and

(xxiv) $\Gamma \vdash T_2[\widetilde{A/X}][v/x] <: T$.

By Lemma 4 with "(ix) and (xix)" and "(xv) and (xxiii)" respectively, we have

- $\Gamma, y : T_2[\widetilde{A/X}][v/x] \vdash C'_{01} <: C'_1[\widetilde{A/X}][v/x]$ and
- $\Gamma, y : T_2[\widetilde{A/X}][v/x] \vdash C_{01} <: C'_{01}$.

Then, by Lemma 16 with these two and (xx), we have

(xxv) $\Gamma, y : T_2[\widetilde{A/X}][v/x] \vdash C_{01} <: C_1[\widetilde{A/X}][v/x]$.

Also, by Lemma 16 with (xxi), (x) and (xvi), we have

(xxvi) $\Gamma \vdash C_2[\widetilde{A/X}][v/x] <: C_{02}$.

From (xvii), (vi), (xxii), (xxiv), (xxv) and (xxvi), we have the conclusion.

5. **Case** (T-LetP)**:** Obvious.

Case (T-LetIp)**:** Contradictory.

Case (T-CSub)**:** We have

(i) $\Gamma \vdash \textbf{let } x = c_1 \textbf{ in } c_2 : \Sigma' \triangleright T' \;/\; S'$,

(ii) $\Gamma \vdash \Sigma' \triangleright T' \;/\; S' <: \Sigma \triangleright T \;/\; \square$, and

(iii) $\Gamma \vdash \Sigma \triangleright T \;/\; \square$

for some Σ', T', and S'. By inversion of (ii), we have

(iv) $S = \square$,

(v) $\Gamma \vdash \Sigma' <: \Sigma$, and

(vi) $\Gamma \vdash T' <: T$.

Then, by the IH of (i), we have

(vii) $\Gamma \vdash c_1 : \Sigma' \triangleright T_1 \;/\; \square$ and

(viii) $\Gamma, x : T_1 \vdash c_2 : \Sigma' \triangleright T' \;/\; \square$

for some T_1. By subsumption of (vii) with (v), we have

(ix) $\Gamma \vdash c_1 : \Sigma \triangleright T_1 \;/\; \square$.

By Lemma 10 with (viii), we have

(x) $\vdash \Gamma, x : T_1$.

Then it holds that $x \notin \mathit{dom}(\Gamma)$ and hence from (iii) we have

(xi) $x \notin \mathit{fv}(T) \cup \mathit{fv}(\Sigma)$.

Also, by Lemma 3 with (ii) and (x), we have

- $\Gamma, x : T_1 \vdash \Sigma' \triangleright T' \;/\; \square <: \Sigma \triangleright T \;/\; \square$.

Then by subsumption of (viii) we have

(xii) $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T \;/\; \square$.

Now we have the conclusion from (ix), (xii) and (xi).

6. **Case** (T-LetP)**:** Contradictory.

Case (T-LetIp)**:** Obvious.

Case (T-CSub)**:** We have

(i) $\Gamma \vdash \textbf{let } x = c_1 \textbf{ in } c_2 : \Sigma' \triangleright T' \;/\; S'$,

(ii) $\Gamma \vdash \Sigma' \triangleright T' \;/\; S' <: \Sigma \triangleright T \;/\; (\forall z.C_1) \Rightarrow C_2$, and

(iii) $\Gamma \vdash \Sigma \triangleright T \;/\; (\forall z.C_1) \Rightarrow C_2$

for some Σ', T', and S'. By inversion of (ii), we have

(iv) $\Gamma \vdash \Sigma' <: \Sigma$,

(v) $\Gamma \vdash T' <: T$, and

(vi) $\Gamma \mid T' \vdash S' <: (\forall z.C_1) \Rightarrow C_2$.

Case analysis on the derivation of (vi).

Case (S-Pure)**:** Contradictory.

Case (S-ATM)**:** We have

(vii) $S' = (\forall z.C_1') \Rightarrow C_2'$,

(viii) $\Gamma, z : T' \vdash C_1 <: C_1'$, and

(ix) $\Gamma \vdash C_2' <: C_2$

for some C_1' and C_2'. Then, by the IH of (i), we have

(x) $\Gamma \vdash c_1 : \Sigma' \triangleright T_1 \;/\; (\forall x.C_0) \Rightarrow C_2'$ and

(xi) $\Gamma, x : T_1 \vdash c_2 : \Sigma' \triangleright T' \;/\; (\forall z.C_1') \Rightarrow C_0$

for some T_1 and C_0. By subsumption of (x) with (iv) and (ix), we have

(xii) $\Gamma \vdash c_1 : \Sigma \triangleright T_1 \;/\; (\forall x.C_0) \Rightarrow C_2$.

By Lemma 10 with (xi), we have

(xiii) $\vdash \Gamma, x : T_1$.

Then it holds that $x \notin \mathit{dom}(\Gamma)$ and hence from (iii) we have

(xiv) $x \notin \mathit{fv}(T) \cup \mathit{fv}(\Sigma) \cup (\mathit{fv}(C_1) \setminus \{z\})$.

Also, by Lemma 3 with (iv), (v), (viii) and (xiii), we have

- $\Gamma, x : T_1 \vdash \Sigma' <: \Sigma$,
- $\Gamma, x : T_1 \vdash T' <: T$, and
- $\Gamma, x : T_1, z : T' \vdash C_1 <: C_1'$.

Then by subsumption of (xi) we have

(xv) $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T \;/\; (\forall z.C_1) \Rightarrow C_0$.

Now we have the conclusion from (xii), (xv) and (xiv).

Case (S-Embed)**:** We have

(xvi) $S' = \Box$,

(xvii) $\Gamma, z : T \vdash C_1 <: C_2$, and

(xviii) $z \notin fv(C_2)$.

Then, by Lemma 18 with (i), we have

(xix) $\Gamma \vdash c_1 : \Sigma' \triangleright T_1 \;/\; \Box$ and

(xx) $\Gamma, x : T_1 \vdash c_2 : \Sigma' \triangleright T' \;/\; \Box$

for some T_1. By Lemma 10 with (xx), we have

(xxi) $\vdash \Gamma, x : T_1$.

Then it holds that $x \notin dom(\Gamma)$ and hence from (iii) we have

(xxii) $x \notin fv(T) \cup fv(\Sigma) \cup (fv(C_1) \setminus \{z\})$ and

(xxiii) $x \notin fv(C_2)$.

Also, by inversion of (iii) we have $\Gamma \vdash C_2$, and so we have $\Gamma \vdash C_2 <: C_2$ by Lemma 15. Then, by Lemma 3 with (xxi) we have $\Gamma, x : T_1 \vdash C_2 <: C_2$. And hence, by (S-Embed) with (xxiii) we have

- $\Gamma \mid T_1 \vdash \Box <: (\forall x.C_2) \Rightarrow C_2$.

Therefore, by subsumption of (xix) with (iv), we have

(xxiv) $\Gamma \vdash c_1 : \Sigma \triangleright T_1 \;/\; (\forall x.C_2) \Rightarrow C_2$.

Moreover, by Lemma 3 with (ii) and (xxi), we have

- $\Gamma, x : T_1 \vdash \Sigma' \triangleright T' \;/\; \Box <: \Sigma \triangleright T \;/\; (\forall z.C_1) \Rightarrow C_2$.

Then by subsumption of (xx) we have

(xxv) $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T \;/\; (\forall z.C_1) \Rightarrow C_2$.

Now we have the conclusion from (xxiv), (xxv) and (xxii).

□

Lemma 19 (Inversion with pure evaluation contexts)**.** *If $\Gamma \vdash K[c] : \Sigma \triangleright T \;/\; (\forall z.C_1) \Rightarrow C_2$, then there exist some y, T_1, and C_0 such that*

- $\Gamma \vdash c : \Sigma \triangleright T_1 \;/\; (\forall y.C_0) \Rightarrow C_2$ *and*
- $\Gamma, y : T_1 \vdash K[\mathbf{return}\ y] : \Sigma \triangleright T \;/\; (\forall z.C_1) \Rightarrow C_0$.

Proof. By induction on the structure of K.

Case $K = [\ ]$**:** We have $\Gamma \vdash c : \Sigma \triangleright T \;/\; (\forall z.C_1) \Rightarrow C_2$. By α-renaming, we have

(i) $\Gamma \vdash c : \Sigma \triangleright T \;/\; (\forall y.C_1[y/z]) \Rightarrow C_2$.

Therefore, we have the first half of the conclusion with $T_1 = T$ and $C_0 = C_1[y/z]$.

On the other hand, from (i), it holds that

(ii) $\vdash \Gamma, y : T$

by Lemma 11, Lemma 10, and inversion. We show the second half of the conclusion by case analysis on T.

Case that T **is a refinement type** $\{z_0 : B \mid \phi\}$**:** By (T-CVar) and (T-Ret) with (ii), it holds that

(iii) $\Gamma, y : T \vdash \mathbf{return}\ y : \emptyset \triangleright \{z_0 : B \mid z_0 = y\} \;/\; \Box$.

Also, we have the following subtyping with Lemma 17:

$$\frac{\overline{\Gamma, y : T, z : \{z_0 : B \mid z_0 = y\} \vdash C_1 <: C_1[y/z]}}{\Gamma, y : T \mid \{z_0 : B \mid z_0 = y\} \vdash \Box <: (\forall z.C_1) \Rightarrow C_1[y/z]}$$

Then it holds that

(iv) $\Gamma, y : T \vdash \emptyset \triangleright \{z_0 : B \mid z_0 = y\} \ / \ \square <: \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C_1[y/z]$

by subtyping. Therefore, by subsumption with (iii) and (iv), we have the conclusion.

Case that T is not a refinement type: By (T-Var) and (T-Ret) with (ii), it holds that

(v) $\Gamma, y : T \vdash \textbf{return}\ y : \emptyset \triangleright T \ / \ \square$.

Also, since T is not a refinement type, by Lemma 8 we have $z \notin C_1$ and so $C_1[y/z] = C_1$. Then, we have the following subtyping with Lemma 15:

$$\frac{\overline{\Gamma, y : T, z : T \vdash C_1 <: C_1[y/z]}}{\Gamma, y : T \mid T \vdash \square <: (\forall z.C_1) \Rightarrow C_1[y/z]}$$

Then it holds that

(vi) $\Gamma, y : T \vdash \emptyset \triangleright T \ / \ \square <: \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C_1[y/z]$

by subtyping. Therefore, by subsumption with (v) and (vi), we have the conclusion.

Case $K = \textbf{let}\ x = K_1\ \textbf{in}\ c_2$: We have $\Gamma \vdash \textbf{let}\ x = K_1[c]\ \textbf{in}\ c_2 : \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C_2$. By Lemma 18, we have

(i) $\Gamma \vdash K_1[c] : \Sigma \triangleright T' \ / \ (\forall x.C') \Rightarrow C_2$,

(ii) $\Gamma, x : T' \vdash c_2 : \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C'$, and

(iii) $x \notin \mathit{fv}(T) \cup \mathit{fv}(\Sigma) \cup (\mathit{fv}(C_1) \setminus \{z\})$

for some T' and C'. By the IH of (i), we have

(iv) $\Gamma \vdash c : \Sigma \triangleright T_1 \ / \ (\forall y.C_0) \Rightarrow C_2$ and

(v) $\Gamma, y : T_1 \vdash K_1[\textbf{return}\ y] : \Sigma \triangleright T' \ / \ (\forall x.C') \Rightarrow C_0$

for some y, T_1 and C_0.

By Lemma 10 with (ii), we have $\vdash \Gamma, x : T'$. By inversion, we have $x \notin \mathit{dom}(\Gamma)$ and $\Gamma \vdash T'$. Also, By Lemma 10 with (v), we have $\vdash \Gamma, y : T_1$. Then, by Lemma 3 we have $\Gamma, y : T_1 \vdash T'$. Moreover, w.l.o.g, we can assume $x \neq y$, and so $x \notin \mathit{dom}(\Gamma) \cup \{y\} = \mathit{dom}(\Gamma, y : T_1)$. Then we have $\vdash \Gamma, y : T_1, x : T'$.

Therefore, by Lemma 3 with (ii), we have

$$\Gamma, y : T_1, x : T' \vdash c_2 : \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C' \ .$$

Then, by (T-LetIp) with (v) and (iii), we have

$$\Gamma, y : T_1 \vdash \textbf{let}\ x = K_1[\textbf{return}\ y]\ \textbf{in}\ c_2 : \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C_0 \ ,$$

that is,

(vi) $\Gamma, y : T_1 \vdash K[\textbf{return}\ y] : \Sigma \triangleright T \ / \ (\forall z.C_1) \Rightarrow C_0$.

Therefore, from (iv) and (vi) we have the conclusion.

□

Theorem 20 (Subject reduction). *If $\emptyset \vdash c : C$ and $c \longrightarrow c'$, then $\emptyset \vdash c' : C$.*

Proof. By induction on the typing derivation.

Case (T-Ret) and (T-Op): Contradictory because there is no evaluation rule for c.

Case (T-App): We have

(i) $c = v_1\ v_2$,

(ii) $C = C_1[v_2/x]$,

(iii) $\vdash v_1 : (x : T_1) \rightarrow C_1$, and

(iv) $\vdash v_2 : T_1$

for some x, v_1, v_2, T_1 and C_1. Case analysis on the evaluation derivation.

Case (E-App): We have

(v) $v_1 = \textbf{rec}(f, x).c_1$, and

(vi) $c' = c_1[v_2/x][(\mathbf{rec}(f,x).c_1)/f]$

for some f, x and c_1. By Lemma 18 with (iii), we have

(vii) $\vdash v_1 : (x : T_0) \to C_0$,

(viii) $\vdash (x : T_0) \to C_0 <: (x : T_1) \to C_1$, and

(ix) $f : (x : T_0) \to C_0, x : T_0 \vdash c : C_0$

for some T_0 and C_0. By Lemma 10 with (ix), inversion, and Lemma 9, we have $\vdash T_0$. Also, by inversion of (viii), we have $\vdash T_1 <: T_0$. Then, By (T-VSub) with (iv), we have $\vdash v_2 : T_0$. Using this and (vii), we have the conclusion by Lemma 5 with (ix).

Case (E-Prim)**:** We have

(x) $v_1 = p$, and

(xi) $c' = \zeta(p, v_2)$

for some p. By Lemma 18 with (iii), we have

(xii) $\vdash p : ty(p)$, and

(xiii) $\vdash ty(p) <: (x : T_1) \to C_1$.

By inversion of (xiii), we have

(xiv) $ty(p) = (x : T_0) \to C_0$,

(xv) $\vdash T_1 <: T_0$, and

(xvi) $x : T_1 \vdash C_0 <: C_1$

for some T_0 and C_0. By Lemma 10 with (xii) and (xiv) and inversion, we have $\vdash T_0$. Then, by (T-VSub) with (iv) and (xv), we have $\vdash v_2 : T_0$. Therefore, by Assumption 2 with (xiv), we have

(xvii) $\vdash \zeta(p, v_2) : C_0[v_2/x]$.

Also, by Lemma 11 with (iii) and inversion, we have

(xviii) $x : T_1 \vdash C_1$.

Using (iv), by Lemma 5 with (xvi) and (xviii) respectively, we have

- $\vdash C_0[v_2/x] <: C_1[v_2/x]$ and
- $\vdash C_1[v_2/x]$.

Therefore, by (T-CSub) with (xvii), we have the conclusion.

Case (T-If)**:** We have

(i) $c = \mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2$,

(ii) $\vdash v : \{x : \mathrm{bool} \mid \phi\}$,

(iii) $v = \mathbf{true} \vdash c_1 : C$, and

(iv) $v = \mathbf{false} \vdash c_2 : C$

for some x, v, c_1, c_2, and ϕ. Case analysis on the evaluation derivation.

Case (E-IfT)**:** We have

(v) $v = \mathbf{true}$, and

(vi) $c' = c_1$.

We have the conclusion by Lemma 14 with (iii).

Case (E-IfF)**:** Similar.

Case (T-CSub)**:** By the IH and (T-CSub).

Case (T-LetP)**:** We have

(i) $c = \mathbf{let}\ x = c_1\ \mathbf{in}\ c_2$,

(ii) $C = \Sigma \triangleright T_2 \ / \ \square$,

(iii) $\vdash c_1 : \Sigma \triangleright T_1 \ / \ \square$,

(iv) $x : T_1 \vdash c_2 : \Sigma \triangleright T_2 \ / \ \square$, and

(v) $x \notin fv(T_2) \cup fv(\Sigma)$

for some x, c_1, c_2, Σ, T_1 and T_2. Case analysis on the evaluation derivation.

Case (E-Let)**:** By the IH and (T-LetP).

Case (E-LetRet)**:** We have

(vi) $c_1 = \textbf{return}\ v$, and

(vii) $c' = c_2[v/x]$

for some v. By Lemma 18 with (iii), we have

(viii) $\vdash T_0 <: T_1$ and

(ix) $\vdash v : T_0$

for some T_0. By Lemma 11 with (iii) and inversion, we have $\vdash T_1$. Then, by (T-VSub) with (ix) and (viii), we have $\vdash v : T_1$. Therefore, by Lemma 5 with (iv), we have

$$\vdash c_2[v/x] : \Sigma \triangleright T_2 \ / \ \square$$

(Note that since (v), it holds that $\Sigma[v/x] = \Sigma$ and $T_2[v/x] = T_2$.) That is, we have the conclusion.

Case (T-LetIp)**:** We have

(i) $c = \textbf{let}\ x = c_1\ \textbf{in}\ c_2$,

(ii) $C = \Sigma \triangleright T_2 \ / \ (\forall z.C_{21}) \Rightarrow C_{12}$,

(iii) $\vdash c_1 : \Sigma \triangleright T_1 \ / \ (\forall x.C_0) \Rightarrow C_{12}$,

(iv) $x : T_1 \vdash c_2 : \Sigma \triangleright T_2 \ / \ (\forall z.C_{21}) \Rightarrow C_0$, and

(v) $x \notin fv(T_2) \cup fv(\Sigma) \cup (fv(C_{21}) \setminus \{z\})$

for some $x, z, c_1, c_2, \Sigma, T_1, T_2, C_0, C_{12}$ and C_{21}. Case analysis on the evaluation derivation.

Case (E-Let)**:** By the IH and (T-LetIp).

Case (E-LetRet)**:** We have

(vi) $c_1 = \textbf{return}\ v$, and

(vii) $c' = c_2[v/x]$

for some v. By Lemma 18 with (iii), we have

(viii) $\vdash T_0 <: T_1$,

(ix) $\vdash v : T_0$, and

(x) $\mid T_0 \vdash \square <: (\forall x.C_0) \Rightarrow C_{12}$

for some T_0. By Lemma 11 with (iii) and inversion, we have $\vdash T_1$. Then, by (T-VSub) with (ix) and (viii), we have $\vdash v : T_1$. Therefore, by Lemma 5 with (iv), we have

(xi) $\vdash c_2[v/x] : \Sigma \triangleright T_2 \ / \ (\forall z.C_{21}) \Rightarrow C_0[v/x]$.

(Note that since (v), it holds that $\Sigma[v/x] = \Sigma$, $T_2[v/x] = T_2$ and $C_{21}[v/x] = C_{21}$.) By inversion of (x), we have

(xii) $x : T_0 \vdash C_0 <: C_{12}$.

By Lemma 11 with (iii) and inversion, we have $\vdash C_{12}$, which means $x \notin fv(C_{12})$. Therefore, by Lemma 5 with (xii), we have

(xiii) $\vdash C_0[v/x] <: C_{12}$.

On the other hand, by Lemma 11 with (iv) and inversion, we have $x : T_1, z : T_2 \vdash C_{21}$. By Lemma 7 with (v), we have $z : T_2 \vdash C_{21}$. Then, by Lemma 15 we have

(xiv) $z : T_2 \vdash C_{21} <: C_{21}$.

Hence, by (S-ATM) with (xiii) and (xiv), we have $\mid T_2 \vdash (\forall z.C_{21}) \Rightarrow C_0[v/x] <: (\forall z.C_{21}) \Rightarrow C_{12}$. Now we have the conclusion by subsumption of (xi).

Case (T-Hndl)**:** We have

(i) $c = \textbf{with}\ h\ \textbf{handle}\ c_0$,

(ii) $h = \{\textbf{return}\ x_r \mapsto c_r, (\mathsf{op}_i(x_i, k_i) \mapsto c_i)_i\}$,

(iii) $\vdash c_0 : \Sigma_0 \triangleright T_0 \ / \ (\forall x_r.C_1) \Rightarrow C$,

(iv) $x_r : T_0 \vdash c_r : C_1$,

(v) $\left(\widetilde{X_i : \widetilde{B}_i}, x_i : T_{i1}, k_i : (y_i : T_{i2}) \to C_{i1} \vdash c_i : C_{i2}\right)_i$, and

(vi) $\Sigma_0 = \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.(x_i : T_{i1}) \to ((y_i : T_{i2}) \to C_{i1}) \to C_{2i})_i\}$

Case analysis on the evaluation derivation.

Case (E-HNDL)**:** By the IH and (T-HNDL).

Case (E-HNDLRET)**:** We have

(vii) $c_0 = \textbf{return } v$ and

(viii) $c' = c_r[v/x_r]$

for some v. By Lemma 18 with (iii), we have

(ix) $\vdash T'_0 <: T_0$,

(x) $\vdash v : T'_0$, and

(xi) $\mid T'_0 \vdash \square <: (\forall x_r.C_1) \Rightarrow C$

for some T'_0. By inversion of (xi), we have

(xii) $x_r : T'_0 \vdash C_1 <: C$ and

(xiii) $x_r \notin \mathit{fv}(C)$.

By Lemma 4 with (iv) and (ix), we have

(xiv) $x_r : T'_0 \vdash c_r : C_1$.

By Lemma 5 with (x) applied to (xii) and (xiv), we have

(xv) $\vdash C_1[v/x_r] <: C$ and

(xvi) $\vdash c_r[v/x_r] : C_1[v/x_r]$

respectively. (Note that $C[v/x_r] = C$ since (xiii).) By Lemma 11 with (iii) and inversion, we have $\vdash C$. From this and (xv) and (xvi), we have the conclusion by (T-CSUB).

Case (E-HNDLOP)**:** We have

(xvii) $c_0 = K[\mathsf{op}_i\ v]$ and

(xviii) $c' = c_i[v/x_i][(\lambda y.\textbf{with } h \textbf{ handle } K[\textbf{return } y])/k_i]$

for some K and v. W.l.o.g., we can assume that y is disjoint from any other existing variables. By Lemma 19 with (iii), we have

(xix) $\vdash \mathsf{op}\ v : \Sigma_0 \triangleright T_1 \ /\ (\forall y.C_0) \Rightarrow C$ and

(xx) $y : T_1 \vdash K[\textbf{return } y] : \Sigma_0 \triangleright T_0 \ /\ (\forall x_r.C_1) \Rightarrow C_0$

for some y, T_1 and C_0. By Lemma 18 with (xix), we have

(xxi) $\Sigma_0 \ni \mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.(x_i : T_{i1}) \to ((y : T_{i2}) \to C_{i1}) \to C_{i2}$,

(xxii) $\vdash \widetilde{A : \widetilde{B}_i}$,

(xxiii) $\vdash v : T_{i1}[\widetilde{A/X_i}]$,

(xxiv) $\vdash T_{i2}[\widetilde{A/X_i}][v/x_i] <: T_1$,

(xxv) $y : T_{i2}[\widetilde{A/X_i}][v/x_i] \vdash C_0 <: C_{i1}[\widetilde{A/X_i}][v/x_i]$, and

(xxvi) $\vdash C_{i2}[\widetilde{A/X_i}][v/x_i] <: C$

for some $\widetilde{A}$. Note that since (vi) holds, it holds that $y = y_i$ and we use $\widetilde{X_i}, \widetilde{\widetilde{B}_i}, x_i, T_{i1}, T_{i2}, C_{i1}$, and C_{i2} here instead of introducing new ones.

Also, by Lemma 4 with (xx) and (xxiv), we have

$$y : T_{i2}[\widetilde{A/X_i}][v/x_i] \vdash K[\textbf{return } y] : \Sigma_0 \triangleright T_0 \ /\ (\forall x_r.C_1) \Rightarrow C_0 .$$

Then, by subsumption with (xxv), we have

(xxvii) $y : T_{i2}[\widetilde{A/X_i}][v/x_i] \vdash K[\textbf{return } y] : \Sigma_0 \triangleright T_0 \ /\ (\forall x_r.C_1) \Rightarrow C_{i1}[\widetilde{A/X_i}][v/x_i]$.

On the other hand, by Lemma 10 with (xxvii) we have $\vdash y : T_{i2}[\widetilde{A/X_i}][v/x_i]$, and hence by Lemma 3 with (iv) and (v), we have

(xxviii) $y : T_{i2}[\widetilde{A/X_i}][v/x_i], x_r : T_0 \vdash c_r : C_1$ and

(xxix) $\left(y : T_{i2}[\widetilde{A/X_i}][v/x_i], \widetilde{X_i : \widetilde{B}_i}, x_i : T_{i1}, k_i : (y_i : T_{i2}) \to C_{i1} \vdash c_i : C_{i2}\right)_i$.

Therefore, by (T-Hndl) with (ii), (vi), (xxvii), (xxviii), and (xxix), we have

$$y : T_{i2}[\widetilde{A/X_i}][v/x_i] \vdash \textbf{with } h \textbf{ handle } K[\textbf{return } y] : C_{i1}[\widetilde{A/X_i}][v/x_i] \ .$$

Then by (T-Fun) we have

$$\text{(xxx)} \ \vdash \lambda y.\textbf{with } h \textbf{ handle } K[\textbf{return } y] : (y : y : T_{i2}[\widetilde{A/X_i}][v/x_i]) \to C_{i1}[\widetilde{A/X_i}][v/x_i] \ .$$

Now, by Lemma 6 with (xxii) applied to (v), we have

$$x_i : T_{i1}[\widetilde{A/X_i}], k_i : (y_i : T_{i2}[\widetilde{A/X_i}]) \to C_{i1}[\widetilde{A/X_i}] \vdash c_i : C_{i2}[\widetilde{A/X_i}] \ .$$

By applying 5 twice with (xxiii) and (xxx) in a row, we have

$$\vdash c_i[v/x_i][(\lambda y.\textbf{with } h \textbf{ handle } K[\textbf{return } y])/k_i] : C_{i2}[\widetilde{A/X_i}][v/x_i] \ .$$

Note that $C_{i2}[\widetilde{A/X_i}][v/x_i][(\lambda y.\textbf{with } h \textbf{ handle } K[\textbf{return } y])/k_i] = C_{i2}[\widetilde{A/X_i}][v/x_i]$ since $k_i \notin fv(C_{i2}[\widetilde{A/X_i}][v/x_i])$ by Lemma 8. Now we have the conclusion by subsumption with (xxvi).

□

4.3 Type Safety

Theorem 21 (Type safety)**.** *If $\emptyset \vdash c : \Sigma \triangleright T \ / \ S$ and $c \longrightarrow^* c'$, then either:*

- $c' = \textbf{return } v$ *for some v such that $\emptyset \vdash v : T$,*
- $c' = K[\mathsf{op}\ v]$ *for some K, op and v such that $\mathsf{op} \in dom(\Sigma)$, or*
- $c' \longrightarrow c''$ *for some c'' such that $\emptyset \vdash c'' : \Sigma \triangleright T \ / \ S$.*

Proof. By induction on the length of $\longrightarrow^*$ with Theorem 13 and Theorem 20. □

5 Definitions for the CPS transformation

5.1 Evaluation rules for the target language of the CPS transformation

$$\text{evaluation context} \quad E ::= [\,] \mid E\ v \mid E\ \widetilde{A} \mid E\ \tau$$

$\boxed{c \longrightarrow c'}$

$$\frac{c \longrightarrow c'}{E[c] \longrightarrow E[c']}\ (\text{Ec-Ctx}) \qquad \frac{}{\textbf{if true then } c_1 \textbf{ else } c_2 \longrightarrow c_1}\ (\text{Ec-IfT}) \qquad \frac{}{\textbf{if false then } c_1 \textbf{ else } c_2 \longrightarrow c_2}\ (\text{Ec-IfF})$$

$$\frac{}{(\textbf{rec}(f : \tau_1, x : \tau_2).c)\ v \longrightarrow c[v/x][(\textbf{rec}(f : \tau_1, x : \tau_2).c)/f]}\ (\text{Ec-App})$$

$$\frac{}{p\ v \longrightarrow \zeta_{cps}(p, v)}\ (\text{Ec-Prim}) \qquad \frac{}{(\Lambda \widetilde{X : \widetilde{B}}.c)\ \widetilde{A} \longrightarrow c[\widetilde{A/X}]}\ (\text{Ec-PApp})$$

$$\frac{}{\{(\mathsf{op}_i = v_i)_i\}\#\mathsf{op}_i \longrightarrow v_i}\ (\text{Ec-Proj}) \qquad \frac{}{(\Lambda \alpha.c)\ \tau \longrightarrow c[\tau/\alpha]}\ (\text{Ec-TApp}) \qquad \frac{}{(c : \tau) \longrightarrow c}\ (\text{Ec-Acsr})$$

5.2 Syntax of typing contexts of the target language of the CPS transformation

$$\Gamma ::= \emptyset \mid \Gamma, x : \tau \mid \Gamma, X : \widetilde{B} \mid \Gamma, \alpha$$

5.3 Well-formedness rules of the target language of the CPS transformation

$\boxed{\vdash \Gamma}$ $\boxed{\Gamma \vdash \tau}$

$$\frac{}{\vdash \emptyset}(\textsc{WEc-Empty}) \quad \frac{\vdash \Gamma \quad x \notin \mathit{dom}(\Gamma) \quad \Gamma \vdash \tau}{\vdash \Gamma, x : \tau}(\textsc{WEc-Var})$$

$$\frac{\vdash \Gamma \quad X \notin \mathit{dom}(\Gamma)}{\vdash \Gamma, X : \widetilde{B}}(\textsc{WEc-PVar}) \quad \frac{\vdash \Gamma \quad \alpha \notin \mathit{dom}(\Gamma)}{\vdash \Gamma, \alpha}(\textsc{WEc-TVar})$$

$$\frac{\Gamma, x : B \vdash \phi}{\Gamma \vdash \{x : B \mid \phi\}}(\textsc{WTc-Rfn}) \quad \frac{\Gamma, x : \tau_1 \vdash \tau_2}{\Gamma \vdash (x : \tau_1) \to \tau_2}(\textsc{WTc-Fun}) \quad \frac{\Gamma, \widetilde{X : \widetilde{B}} \vdash \tau}{\Gamma \vdash \forall \widetilde{X : \widetilde{B}}.\tau}(\textsc{WTc-PPoly})$$

$$\frac{(\Gamma \vdash \tau_i)_i}{\Gamma \vdash \{(\mathsf{op}_i : \tau_i)_i\}}(\textsc{WTc-Rcd}) \quad \frac{\alpha \in \Gamma}{\Gamma \vdash \alpha}(\textsc{WTc-TVar}) \quad \frac{\Gamma, \alpha \vdash \tau}{\Gamma \vdash \forall \alpha.\tau}(\textsc{WTc-TPoly})$$

5.4 Typing rules of the target language of the CPS transformation

$\boxed{\Gamma \vdash c : \tau}$

$$\frac{\vdash \Gamma \quad \Gamma(x) = \{y : B \mid \phi\}}{\Gamma \vdash x : \{y : B \mid x = y\}}(\textsc{Tc-CVar}) \quad \frac{\vdash \Gamma \quad \forall y, B, \phi.\Gamma(x) \neq \{y : B \mid \phi\}}{\Gamma \vdash x : \Gamma(x)}(\textsc{Tc-Var}) \quad \frac{\vdash \Gamma}{\Gamma \vdash p : \mathit{ty}_{\mathit{cps}}(p)}(\textsc{Tc-Prim})$$

$$\frac{\Gamma, f : (x : \tau_1) \to \tau_2, x : \tau_1 \vdash c : \tau_2}{\Gamma \vdash \mathbf{rec}(f : (x : \tau_1) \to \tau_2, x : \tau_1).c : (x : \tau_1) \to \tau_2}(\textsc{Tc-Fun}) \quad \frac{\Gamma \vdash c : (x : \tau_1) \to \tau_2 \quad \Gamma \vdash v : \tau_1}{\Gamma \vdash c\ v : \tau_2[v/x]}(\textsc{Tc-App})$$

$$\frac{\Gamma, \alpha \vdash c : \tau}{\Gamma \vdash \Lambda \alpha.c : \forall \alpha.\tau}(\textsc{Tc-TAbs}) \quad \frac{\Gamma \vdash c : \forall \alpha.\tau' \quad \Gamma \vdash \tau}{\Gamma \vdash c\ \tau : \tau'[\tau/\alpha]}(\textsc{Tc-TApp})$$

$$\frac{\Gamma, \widetilde{X : \widetilde{B}} \vdash c : \tau}{\Gamma \vdash \Lambda \widetilde{X : \widetilde{B}}.c : \forall \widetilde{X : \widetilde{B}}.\tau}(\textsc{Tc-PAbs}) \quad \frac{\Gamma \vdash c : \forall \widetilde{X : \widetilde{B}}.\tau \quad \Gamma \vdash \widetilde{A : \widetilde{B}}}{\Gamma \vdash c\ \widetilde{A} : \tau[\widetilde{A/X}]}(\textsc{Tc-PApp})$$

$$\frac{(\Gamma \vdash v_i : \tau_i)_i}{\Gamma \vdash \{(\mathsf{op}_i = v_i)_i\} : \{(\mathsf{op}_i : \tau_i)_i\}}(\textsc{Tc-Rcd}) \quad \frac{\Gamma \vdash v : \{(\mathsf{op}_i : \tau_i)_i\}}{\Gamma \vdash v\#\mathsf{op}_i : \tau_i}(\textsc{Tc-Proj})$$

$$\frac{\Gamma \vdash v : \{x : \mathrm{bool} \mid \phi\} \quad \Gamma, v = \mathbf{true} \vdash c_1 : \tau \quad \Gamma, v = \mathbf{false} \vdash c_2 : \tau}{\Gamma \vdash \mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2 : \tau}(\textsc{Tc-If})$$

$$\frac{\Gamma \vdash c : \tau' \quad \Gamma \vdash \tau' <: \tau \quad \Gamma \vdash \tau}{\Gamma \vdash (c : \tau) : \tau}(\textsc{Tc-Ascr}) \quad \frac{\Gamma \vdash c : \tau_1 \quad \Gamma \vdash \tau_1 <: \tau_2 \quad \Gamma \vdash \tau_2}{\Gamma \vdash c : \tau_2}(\textsc{Tc-Sub})$$

5.5 Subtyping rules of the target language of the CPS transformation

$\boxed{\Gamma \vdash \tau_1 <: \tau_2}$

$$\frac{\Gamma, x : B \vDash \phi_1 \implies \phi_2}{\Gamma \vdash \{x : B \mid \phi_1\} <: \{x : B \mid \phi_2\}}(\textsc{Sc-Rfn}) \quad \frac{\Gamma \vdash \tau_{21} <: \tau_{11} \quad \Gamma, x : \tau_{21} \vdash \tau_{12} <: \tau_{22}}{\Gamma \vdash (x : \tau_{11}) \to \tau_{12} <: (x : \tau_{21}) \to \tau_{22}}(\textsc{Sc-Fun})$$

$$\frac{\Gamma, \widetilde{X : \widetilde{B}} \vdash \tau_1 <: \tau_2}{\Gamma \vdash \forall \widetilde{X : \widetilde{B}}.\tau_1 <: \forall \widetilde{X : \widetilde{B}}.\tau_2}(\textsc{Sc-PPoly}) \quad \frac{(\Gamma \vdash \tau_{1i} <: \tau_{2i})_i}{\Gamma \vdash \{(\mathsf{op}_i : \tau_{1i})_i, (\mathsf{op}'_i : \tau'_i)_i\} <: \{(\mathsf{op}_i : \tau_{2i})_i\}}(\textsc{Sc-Rcd})$$

$$\frac{\alpha \in \Gamma}{\Gamma \vdash \alpha <: \alpha}(\textsc{Sc-TVar}) \quad \frac{\Gamma, \beta \vdash \tau_1[\tau/\alpha] <: \tau_2 \quad \Gamma, \beta \vdash \tau \quad \beta \notin \mathit{fv}(\forall \alpha.\tau_1)}{\Gamma \vdash \forall \alpha.\tau_1 <: \forall \beta.\tau_2}(\textsc{Sc-Poly})$$

5.6 CPS transformation of expressions

$$\begin{aligned}
[\![x]\!] &\stackrel{\text{def}}{=} x \\
[\![p]\!] &\stackrel{\text{def}}{=} \mathit{cps}(p) \\
[\![\mathbf{rec}(f^{(x:T_1) \to C_1}, x^{T_2}).c]\!] &\stackrel{\text{def}}{=} \mathbf{rec}(f : [\![(x : T_1) \to C_1]\!], x : [\![T_2]\!]).[\![c]\!] \\
[\![\mathbf{return}\ v^T]\!] &\stackrel{\text{def}}{=} \overline{\Lambda}\alpha.\overline{\lambda}h : \{\}.\overline{\lambda}k : [\![T]\!] \to \alpha.k\ [\![v]\!]
\end{aligned}$$

$$\begin{aligned}
\llbracket \mathbf{let}\ x = c_1^{\Sigma \triangleright T_1/\square}\ \mathbf{in}\ c_2^{\Sigma \triangleright T_2/\square} \rrbracket &\overset{\text{def}}{=} \overline{\Lambda}\alpha.\overline{\lambda} h : \llbracket \Sigma \rrbracket.\overline{\lambda} k : \llbracket T_2 \rrbracket \to \alpha.\llbracket c_1 \rrbracket \mathbin{\overline{@}} \alpha \mathbin{\overline{@}} h \mathbin{\overline{@}} (\lambda x : \llbracket T_1 \rrbracket.\llbracket c_2 \rrbracket \mathbin{\overline{@}} \alpha \mathbin{\overline{@}} h \mathbin{\overline{@}} k) \\
\llbracket \mathbf{let}\ x = c_1^{\Sigma \triangleright T_1/(\forall x.C_1) \Rightarrow C_2}\ \mathbf{in}\ c_2^{\Sigma \triangleright T_2/(\forall z.C_0) \Rightarrow C_1} \rrbracket &\overset{\text{def}}{=} \\
\overline{\Lambda}\alpha.\overline{\lambda} h : \llbracket \Sigma \rrbracket.\overline{\lambda} k : (z : \llbracket T_2 \rrbracket) \to \llbracket C_0 \rrbracket.\llbracket c_1 \rrbracket &\mathbin{\overline{@}} \llbracket C_2 \rrbracket \mathbin{\overline{@}} h \mathbin{\overline{@}} (\lambda x : \llbracket T_1 \rrbracket.\llbracket c_2 \rrbracket \mathbin{\overline{@}} \llbracket C_1 \rrbracket \mathbin{\overline{@}} h \mathbin{\overline{@}} k) \\
\llbracket v_1\ v_2 \rrbracket &\overset{\text{def}}{=} \llbracket v_1 \rrbracket\ \llbracket v_2 \rrbracket \\
\llbracket (\mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2)^C \rrbracket &\overset{\text{def}}{=} (\mathbf{if}\ \llbracket v \rrbracket\ \mathbf{then}\ \llbracket c_1 \rrbracket\ \mathbf{else}\ \llbracket c_2 \rrbracket : \llbracket C \rrbracket) \\
\llbracket (\mathsf{op}^{\widetilde{A}}\ v)^{\Sigma \triangleright T/(\forall y.C_1) \Rightarrow C_2} \rrbracket &\overset{\text{def}}{=} \overline{\Lambda}\alpha.\overline{\lambda} h : \llbracket \Sigma \rrbracket.\overline{\lambda} k : (y : \llbracket T \rrbracket \to \llbracket C_1 \rrbracket).h\#\mathsf{op}\ \widetilde{A}\ \llbracket v \rrbracket\ (\lambda y' : \llbracket T \rrbracket.k\ y') \\
\llbracket (\mathbf{with}\ h\ \mathbf{handle}\ c)^C \rrbracket &\overset{\text{def}}{=} \llbracket c \rrbracket \mathbin{\overline{@}} \llbracket C \rrbracket \mathbin{\overline{@}} \llbracket h^{ops} \rrbracket \mathbin{\overline{@}} \llbracket h^{ret} \rrbracket \\
&\text{where} \begin{cases} h = \{\mathbf{return}\ x_r^{T_r} \mapsto c_r, (\mathsf{op}_i^{\widetilde{X_i : \widetilde{B}_i}}(x_i^{T_{x_i}}, k_i^{T_{k_i}}) \mapsto c_i)_i\} \\ \llbracket h^{ops} \rrbracket \overset{\text{def}}{=} \{(\mathsf{op}_i = \Lambda \widetilde{X_i : \widetilde{B}_i}.\lambda x_i : \llbracket T_{x_i} \rrbracket.\lambda k_i : \llbracket T_{k_i} \rrbracket.\llbracket c_i \rrbracket)_i\} \\ \llbracket h^{ret} \rrbracket \overset{\text{def}}{=} \lambda x_r : \llbracket T_r \rrbracket.\llbracket c_r \rrbracket \end{cases}
\end{aligned}$$

5.7 CPS transformation of types and typing contexts

$$\begin{aligned}
\llbracket \{x : B \mid \phi\} \rrbracket &\overset{\text{def}}{=} \{x : B \mid \phi\} \\
\llbracket (x : T) \to C \rrbracket &\overset{\text{def}}{=} (x : \llbracket T \rrbracket) \to \llbracket C \rrbracket \\
\llbracket \Sigma \triangleright T\ /\ (\forall x.C_1) \Rightarrow C_2 \rrbracket &\overset{\text{def}}{=} \forall _.\llbracket \Sigma \rrbracket \to ((x : \llbracket T \rrbracket) \to \llbracket C_1 \rrbracket) \to \llbracket C_2 \rrbracket \\
\llbracket \Sigma \triangleright T\ /\ \square \rrbracket &\overset{\text{def}}{=} \forall \alpha.\llbracket \Sigma \rrbracket \to (\llbracket T \rrbracket \to \alpha) \to \alpha \\
\llbracket \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.F_i)_i\} \rrbracket &\overset{\text{def}}{=} \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.\llbracket F_i \rrbracket^{\mathcal{F}})_i\} \\
\llbracket (x : T_1) \to ((y : T_2) \to C_1) \to C_2 \rrbracket^{\mathcal{F}} &\overset{\text{def}}{=} (x : \llbracket T_1 \rrbracket) \to \llbracket ((y : T_2) \to C_1) \rrbracket \to \llbracket C_2 \rrbracket
\end{aligned}$$

$$\begin{aligned}
\llbracket \emptyset \rrbracket &\overset{\text{def}}{=} \emptyset \\
\llbracket \Gamma, x : T \rrbracket &\overset{\text{def}}{=} \llbracket \Gamma \rrbracket, x : \llbracket T \rrbracket \\
\llbracket \Gamma, X : \widetilde{B} \rrbracket &\overset{\text{def}}{=} \llbracket \Gamma \rrbracket, X : \widetilde{B}
\end{aligned}$$

6 Proof of dynamic semantics preservation of the CPS transformation

Regarding dynamic semantics, we identify values and computations modulo types and predicates since they are irrelevant to the dynamic semantics. That is, the following equations hold, for example:

$$\begin{gathered}
\lambda x : \tau_1.c = \lambda x : \tau_2.c \\
c[\tau/\alpha] = c \\
c[A/X] = c
\end{gathered}$$

Also, We often omit type annotations when they are unnecessary.

Moreover, We also identify values and computations modulo β equivalence of the (static) meta language (this is admissible because the meta language is pure). Formally, we define a relation $\equiv_\beta$ as the smallest congruence relation over expressions in the target language that satisfies the following equations:

$$\begin{aligned}
(\overline{\lambda} x : \tau.c) \mathbin{\overline{@}} v &\equiv_\beta c[v/x] \\
(\overline{\Lambda}\alpha.c) \mathbin{\overline{@}} \tau &\equiv_\beta c[\tau/\alpha]
\end{aligned}$$

and we admit the $\equiv_\beta$-equivalence.

Assumption 22.

- $cps(p)\ \llbracket v \rrbracket \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \longrightarrow^* \llbracket \zeta(p, v) \rrbracket \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k$
- If $\zeta(p, v)$ is undefined, then $cps(p)\ \llbracket v \rrbracket$ gets stuck.

- $p = \textbf{true} \iff cps(p) = \textbf{true}$
- $p = \textbf{false} \iff cps(p) = \textbf{false}$

Lemma 23 (CPS transformation is homomorphic for substitution)**.**

- $[\![v[v_0/x]]\!] = [\![v]\!][[\![v_0]\!]/x]$
- $[\![c[v_0/x]]\!] = [\![c]\!][[\![v_0]\!]/x]$

Proof. By simultaneous induction on the structure of v and c. ☐

Lemma 24 (Evaluation with pure evaluation context)**.**

$$[\![K[\mathsf{op}\ v]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \longrightarrow^* v_h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.[\![K[\textbf{return}\ y]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k)\ .$$

Proof. By induction on the structure of K.

Case $K = [\ ]$**:**

$$\begin{aligned}
\text{LHS} &= [\![\mathsf{op}\ v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.k\ y)) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&\longrightarrow^* v_h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.v_k\ y) \\
&\equiv_\beta v_h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.(\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.k\ y) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&= v_h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.[\![\textbf{return}\ y]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&= \text{RHS}
\end{aligned}$$

Case $K = \textbf{let}\ x = K_1\ \textbf{in}\ c_2$**:**

$$\begin{aligned}
\text{LHS} &= [\![\textbf{let}\ x = K_1[\mathsf{op}\ v]\ \textbf{in}\ c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.[\![K_1[\mathsf{op}\ v]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} k)) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&\longrightarrow^* [\![K_1[\mathsf{op}\ v]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&\text{(by the IH)} \\
&\longrightarrow^* v_h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.[\![K_1[\textbf{return}\ y]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k)) \\
&\equiv_\beta v_h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.(\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.[\![K_1[\textbf{return}\ y]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} k)) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&= v_h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.[\![\textbf{let}\ x = K_1[\textbf{return}\ y]\ \textbf{in}\ c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&= v_h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y.[\![K[\textbf{return}\ y]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&= \text{RHS}
\end{aligned}$$

☐

Lemma 25 (One-step simulation)**.** *If* $c \longrightarrow c'$*, then* $[\![c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \longrightarrow^* [\![c']\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k$.

Proof. By induction on the derivation of $c \longrightarrow c'$. In the following, we implicitly use Lemma 23 and the equality $c[\tau/\alpha] = c$ and $c[A/X] = c$ (note that we identify computations modulo types and predicates regarding dynamic semantics).

Case (E-Let)**:**

$$\begin{aligned}
\text{LHS} &= [\![\textbf{let}\ x = c_1\ \textbf{in}\ c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.[\![c_1]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} k)) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&\longrightarrow^* [\![c_1]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&\text{(by the IH)} \\
&\longrightarrow^* [\![c_1']\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&\equiv_\beta (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.[\![c_1']\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} k)) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= [\![\textbf{let}\ x = c_1'\ \textbf{in}\ c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= \text{RHS}
\end{aligned}$$

Case (E-LetRet)**:** First, w.l.o.g., we can assume that $x \notin fv(v_h) \cup fv(v_k)$. Then,

$$\begin{aligned}
\text{LHS} &= [\![\textbf{let } x = \textbf{return } v \textbf{ in } c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.[\![\textbf{return } v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} h \mathbin{\overline{@}} k)) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&\longrightarrow^* [\![\textbf{return } v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.k\ [\![v]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k) \\
&\longrightarrow^* (\lambda x.[\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k)\ [\![v]\!] \\
&\longrightarrow ([\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k)[[\![v]\!]/x] \\
&= ([\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k)[[\![v]\!]/x] \\
&= [\![c_2]\!][[\![v]\!]/x] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= [\![c_2[v/x]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= \text{RHS}
\end{aligned}$$

Case (E-IfT)**:**

$$\begin{aligned}
\text{LHS} &= [\![(\textbf{if true then } c_1 \textbf{ else } c_2)^C]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= (\textbf{if true then } [\![c_1]\!] \textbf{ else } [\![c_2]\!] : [\![C]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&\longrightarrow^* [\![c_1]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= \text{RHS}
\end{aligned}$$

Case (E-IfF)**:** similar.

Case (E-App)**:**

$$\begin{aligned}
\text{LHS} &= [\![(\textbf{rec}(f,x).c)\ v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= (\textbf{rec}(f,x).[\![c]\!])\ [\![v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&\longrightarrow [\![c]\!][\textbf{rec}(f,x).[\![c]\!]/f, [\![v]\!]/x] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= [\![c[\textbf{rec}(f,x).c/f, v/x]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= \text{RHS}
\end{aligned}$$

Case (E-Prim)**:** By Assumption 22.

Case (E-Hndl)**:**

$$\begin{aligned}
\text{LHS} &= [\![\textbf{with } h \textbf{ handle } c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= ([\![c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} [\![h^{ops}]\!] \mathbin{\overline{@}} [\![h^{ret}]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&\text{(by the IH)} \\
&= ([\![c']\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} [\![h^{ops}]\!] \mathbin{\overline{@}} [\![h^{ret}]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= [\![\textbf{with } h \textbf{ handle } c']\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= \text{RHS}
\end{aligned}$$

Case (E-HndlRet)**:**

$$\begin{aligned}
\text{LHS} &= [\![\textbf{with } h \textbf{ handle return } v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= ([\![\textbf{return } v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} [\![h^{ops}]\!] \mathbin{\overline{@}} [\![h^{ret}]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= (((\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.k\ [\![v]\!])) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} [\![h^{ops}]\!] \mathbin{\overline{@}} [\![h^{ret}]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&\longrightarrow^* ([\![h^{ret}]\!]\ [\![v]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= ((\lambda x_r.[\![c_r]\!])\ [\![v]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= [\![c_r]\!][[\![v]\!]/x_r] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= [\![c_r[v/x_r]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\
&= \text{RHS}
\end{aligned}$$

Case (E-HndlOp)**:**

$$\begin{aligned}\text{LHS} &= [\![\textbf{with } h \textbf{ handle } K[\mathsf{op}_i\ v]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\ &= ([\![K[\mathsf{op}_i\ v]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} [\![h^{ops}]\!] \mathbin{\overline{@}} [\![h^{ret}]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\ &\quad \text{(by Lemma 24)} \\ &\longrightarrow^* [\![h^{ops}]\!]\#\mathsf{op}_i\ \widetilde{A}\ [\![v]\!]\ (\lambda y.[\![K[\textbf{return } y]]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} [\![h^{ops}]\!] \mathbin{\overline{@}} [\![h^{ret}]\!]) \\ &= [\![h^{ops}]\!]\#\mathsf{op}_i\ \widetilde{A}\ [\![v]\!]\ (\lambda y.[\![\textbf{with } h \textbf{ handle } K[\textbf{return } y]]\!]) \\ &\longrightarrow (\Lambda \widetilde{X_i}.\lambda x_i.\lambda k_i.[\![c_i]\!])\ \widetilde{A}\ [\![v]\!]\ (\lambda y.[\![\textbf{with } h \textbf{ handle } K[\textbf{return } y]]\!]) \\ &\longrightarrow^* [\![c_i]\!][[\![v]\!]/x_i][\lambda y.[\![\textbf{with } h \textbf{ handle } K[\textbf{return } y]]\!]/k_i] \\ &= [\![c_i[v/x_i][\lambda y.\textbf{with } h \textbf{ handle } K[\textbf{return } y]/k_i]]\!] \\ &= \text{RHS}\end{aligned}$$

□

Theorem 26 (Forward (multi-step) simulation)**.** *If* $c \longrightarrow^* \textbf{return } v$*, then* $[\![c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} \{\} \mathbin{\overline{@}} (\lambda x : \tau.x) \longrightarrow^+ [\![v]\!]$ *.*

Proof. By applying Lemma 25 repeatedly, we have

$$[\![c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} \{\} \mathbin{\overline{@}} (\lambda x : \tau.x) \longrightarrow^* [\![\textbf{return } v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} \{\} \mathbin{\overline{@}} (\lambda x : \tau.x)\ .$$

Then,

$$\begin{aligned}&[\![\textbf{return } v]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} \{\} \mathbin{\overline{@}} (\lambda x : \tau.x) \\ &= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.k\ [\![v]\!]) \mathbin{\overline{@}} \tau \mathbin{\overline{@}} \{\} \mathbin{\overline{@}} (\lambda x : \tau.x) \\ &\longrightarrow^* (\lambda x.x)\ [\![v]\!] \\ &\longrightarrow [\![v]\!]\end{aligned}$$

and therefore we have the conclusion. □

Definition 27. We define evaluation contexts E as follows:

$$E ::= [\ ] \mid \textbf{let } x = E \textbf{ in } c \mid \textbf{with } h \textbf{ handle } E$$

Definition 28. We define a function *bop* as follows:

$$\begin{aligned} bop([\ ]) &\stackrel{\text{def}}{=} \emptyset \\ bop(\textbf{let } x = E \textbf{ in } c) &\stackrel{\text{def}}{=} bop(E) \\ bop(\textbf{with } h \textbf{ handle } E) &\stackrel{\text{def}}{=} dom(h) \cup bop(E)\end{aligned}$$

That is, $bop(E)$ is a set of operations that are handled by a handler in E.

We say c *is stuck* if c is irreducible and $c \neq \textbf{return } v$. We proceed the proof of the backward simulation following **?**.

Lemma 29 (Preservation of the specific forms of stuck computations)**.**

1. *If* $c = E[\textbf{if } v \textbf{ then } c_1 \textbf{ else } c_2]$ *where* v *is not* **true** *nor* **false***, then* $[\![c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k$ *gets stuck.*
2. *If* $c = E[v_1\ v_2]$ *where* v_1 *is not* $\textbf{rec}(f, x).c$ *nor* p *such that* $\zeta(p, v_2)$ *is defined, then* $[\![c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k$ *gets stuck.*
3. *Let* h_0 *be a handler. If* $c = E[\mathsf{op}\ v]$ *where* $\mathsf{op} \notin bop(E) \cup dom(h_0)$*, then* $[\![c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} [\![h_0^{\mathsf{ops}}]\!] \mathbin{\overline{@}} v_k$ *gets stuck.*

Proof.

1. By induction on the structure of E.

   Case $E = [\ ]$**:**

$$\begin{aligned}[\![c]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k &= [\![\textbf{if } v \textbf{ then } c_1 \textbf{ else } c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k \\ &= \textbf{if } [\![v]\!] \textbf{ then } [\![c_1]\!] \textbf{ else } [\![c_2]\!] \mathbin{\overline{@}} \tau \mathbin{\overline{@}} v_h \mathbin{\overline{@}} v_k\end{aligned}$$

   From Assumption 22, $[\![v]\!]$ is neither **true** nor **false**. Therefore, there is no applicable evaluation rule, and hence this computation is stuck.

Case $E = \textbf{let } x = E_1 \textbf{ in } c$**:**

$$\begin{aligned}\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, v_h \,\overline{@}\, v_k &= \llbracket \textbf{let } x = E_1[\textbf{if } v \textbf{ then } c_1 \textbf{ else } c_2] \textbf{ in } c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, v_h \,\overline{@}\, v_k \\ &= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.\llbracket E_1[\textbf{if } v \textbf{ then } c_1 \textbf{ else } c_2] \rrbracket \,\overline{@}\, \tau \,\overline{@}\, h \,\overline{@}\, (\lambda x.\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, h \,\overline{@}\, k))\,\overline{@}\, \tau \,\overline{@}\, v_h \,\overline{@}\, v_k \\ &\longrightarrow^* \llbracket E_1[\textbf{if } v \textbf{ then } c_1 \textbf{ else } c_2] \rrbracket \,\overline{@}\, \tau \,\overline{@}\, v_h \,\overline{@}\, (\lambda x.\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, v_h \,\overline{@}\, v_k)\end{aligned}$$

By the IH, this computation gets stuck.

Case $E = \textbf{with } h \textbf{ handle } E_1$**:**

$$\begin{aligned}\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, v_h \,\overline{@}\, v_k &= \llbracket \textbf{with } h \textbf{ handle } E_1[\textbf{if } v \textbf{ then } c_1 \textbf{ else } c_2] \rrbracket \,\overline{@}\, \tau \,\overline{@}\, v_h \,\overline{@}\, v_k \\ &= (\llbracket E_1[\textbf{if } v \textbf{ then } c_1 \textbf{ else } c_2] \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h^{ops} \rrbracket \,\overline{@}\, \llbracket h^{ret} \rrbracket)\,\overline{@}\, \tau \,\overline{@}\, v_h \,\overline{@}\, v_k\end{aligned}$$

By the IH, this computation gets stuck.

2. Similar to the case 1.

3. By induction on the structure of E.

Case $E = [\,]$**:**

$$\begin{aligned}\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k &= \llbracket \mathsf{op}\ v \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k \\ &= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.h\#\mathsf{op}\ \widetilde{A}\ \llbracket v \rrbracket\ (\lambda y.\llbracket \textbf{return } y \rrbracket \,\overline{@}\, \tau \,\overline{@}\, h \,\overline{@}\, k))\,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k \\ &\longrightarrow^* h_0^{\mathsf{ops}}\#\mathsf{op}\ \widetilde{A}\ \llbracket v \rrbracket\ (\lambda y.\llbracket \textbf{return } y \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k)\end{aligned}$$

Here, h_0^{ops} does not have a field with op since $\mathsf{op} \notin \mathit{dom}(h_0)$. Therefore, there is no applicable evaluation rule, and hence this computation is stuck.

Case $E = \textbf{let } x = E_1 \textbf{ in } c$**:**

$$\begin{aligned}\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k &= \llbracket \textbf{let } x = E_1[\mathsf{op}\ v] \textbf{ in } c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k \\ &= (\overline{\Lambda}\alpha.\overline{\lambda}h.\overline{\lambda}k.\llbracket E_1[\mathsf{op}\ v] \rrbracket \,\overline{@}\, \tau \,\overline{@}\, h \,\overline{@}\, (\lambda x.\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, h \,\overline{@}\, k))\,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k \\ &\longrightarrow^* \llbracket E_1[\mathsf{op}\ v] \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, (\lambda x.\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k)\end{aligned}$$

Since $\mathsf{op} \notin \mathit{bop}(E) \cup \mathit{dom}(h_0)$ and $\mathit{bop}(E) = \mathit{bop}(\textbf{let } x = E_1 \textbf{ in } c) = \mathit{bop}(E_1)$, it holds that $\mathsf{op} \notin \mathit{bop}(E_1) \cup \mathit{dom}(h_0)$. Then, by the IH, this computation gets stuck.

Case $E = \textbf{with } h \textbf{ handle } E_1$**:**

$$\begin{aligned}\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k &= \llbracket \textbf{with } h \textbf{ handle } E_1[\mathsf{op}\ v] \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k \\ &= (\llbracket E_1[\mathsf{op}\ v] \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \llbracket h^{ops} \rrbracket \,\overline{@}\, \llbracket h^{ret} \rrbracket)\,\overline{@}\, \tau \,\overline{@}\, \llbracket h_0^{\mathsf{ops}} \rrbracket \,\overline{@}\, v_k\end{aligned}$$

Here, $\mathsf{op} \notin \mathit{bop}(E) = \mathit{bop}(\textbf{with } h \textbf{ handle } E_1) = \mathit{bop}(E_1) \cup \mathit{dom}(h)$. Therefore, by the IH, this computation gets stuck.

□

Lemma 30 (Preservation of stuck computations)**.** *If c is a stuck computation, then $\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \{\} \,\overline{@}\, (\lambda x : \tau.x)$ also gets stuck.*

Proof. A stuck computation c is either:

- $E[\textbf{if } v \textbf{ then } c_1 \textbf{ else } c_2]$ where v is not **true** nor **false**,
- $E[v_1\ v_2]$ where v_1 is not $\textbf{rec}(f, x).c$ nor p such that $\zeta(p, v_2)$ is defined, or
- $E[\mathsf{op}\ v]$ where $\mathsf{op} \notin \mathit{bop}(E)$.

Therefore, it is immediate from Lemma 29. □

Theorem 31 (Backward simulation)**.** *If $\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \{\} \,\overline{@}\, (\lambda x : \tau.x) \longrightarrow^+ v'$, then $c \longrightarrow^* \textbf{return } v$ and $\llbracket v \rrbracket = v'$.*

Proof. We show this theorem by proving its contraposition: If "$c \longrightarrow^* \textbf{return } v$ and $\llbracket v \rrbracket = v'$" does not hold, then $\llbracket c \rrbracket \,\overline{@}\, \tau \,\overline{@}\, \{\} \,\overline{@}\, (\lambda x : \tau.x) \longrightarrow^+ v'$ also does not hold. We can divide the situation into two cases:

Case that $c \longrightarrow^* \textbf{return } v$ does not hold: There are two possibilities where c does not evaluate to a value-return.

Case that c diverges: Since c diverges, for all natural numbers n, there exists a sequence

$$c \longrightarrow c_1 \longrightarrow \cdots \longrightarrow c_n \ .$$

Then, by Lemma 25, we have a sequence

$$[\![c]\!] \ \overline{@} \ \tau \ \overline{@} \ \{\} \ \overline{@} \ (\lambda x : \tau.x) \longrightarrow^+ [\![c_1]\!] \ \overline{@} \ \tau \ \overline{@} \ \{\} \ \overline{@} \ (\lambda x : \tau.x) \longrightarrow^+ \cdots \longrightarrow^+ [\![c_n]\!] \ \overline{@} \ \tau \ \overline{@} \ \{\} \ \overline{@} \ (\lambda x : \tau.x)$$

for all n. The length of the sequence is at least n, and therefore $[\![c]\!] \overline{@} \tau \overline{@} \{\} \overline{@} (\lambda x : \tau.x)$ has evaluation sequences of arbitrary length, which means it cannot be evaluated to a value.

Case that $c \longrightarrow^* c'$ and c' is stuck: By applying Lemma 25 repeatedly, we have

$$[\![c]\!] \ \overline{@} \ \tau \ \overline{@} \ \{\} \ \overline{@} \ (\lambda x : \tau.x) \longrightarrow^* [\![c']\!] \ \overline{@} \ \tau \ \overline{@} \ \{\} \ \overline{@} \ (\lambda x : \tau.x) \ .$$

Also, by Lemma 30, it holds that $[\![c']\!] \overline{@} \tau \overline{@} \{\} \overline{@} (\lambda x : \tau.x)$ gets stuck. Therefore, $[\![c]\!] \overline{@} \tau \overline{@} \{\} \overline{@} (\lambda x : \tau.x)$ cannot be evaluated to a value.

Case that $c \longrightarrow^*$ return v holds but $[\![v]\!] = v'$ does not: By Theorem 26, we have

$$[\![c]\!] \ \overline{@} \ \tau \ \overline{@} \ \{\} \ \overline{@} \ (\lambda x : \tau.x) \longrightarrow^+ [\![v]\!] \ .$$

Then, from the premise $[\![v]\!] \neq v'$ and the fact that the evaluation of the target language is deterministic, it cannot be the case that $[\![c]\!] \ \overline{@} \ \tau \ \overline{@} \ \{\} \ \overline{@} \ (\lambda x : \tau.x) \longrightarrow^+ v'$.

□

Corollary 32 (Simulation)**.** *If $c \longrightarrow^*$ **return** v, then $[\![c]\!] \overline{@} \tau \overline{@} \{\} \overline{@} (\lambda x : \tau.x) \longrightarrow^+ [\![v]\!]$. Also, if $[\![c]\!] \overline{@} \tau \overline{@} \{\} \overline{@} (\lambda x : \tau.x) \longrightarrow^+ v'$, then $c \longrightarrow^*$ **return** v and $[\![v]\!] = v'$.*

Proof. Immediate from Theorem 26 and 31. □

7 Proof of type preservation of the CPS transformation

In the following, we consider static expressions and dynamic ones as identical since the distinction is irrelevant to the discussion on the type preservation. In other words, we write $[\![c]\!] \ \overline{@} \ \tau \ \overline{@} \ v_h \ \overline{@} \ v_k$ as $[\![c]\!] \ \tau \ v_h \ v_k$ below, for example.

7.1 Basic properties for the target language of the CPS transformation

Assumption 33.

- If $\vdash \Gamma$ and $dom(\Gamma) \supseteq fv(\phi)$, then $\Gamma \vdash \phi$.
- If $\Gamma \vdash \phi$, then $\vdash \Gamma$.
- If $\Gamma \vdash \phi$, then $\Gamma \vDash \phi \Rightarrow \phi$.
- If $\Gamma \vDash \phi_1 \Rightarrow \phi_2$ and $\Gamma \vDash \phi_2 \Rightarrow \phi_3$, then $\Gamma \vDash \phi_1 \Rightarrow \phi_3$.
- If $\Gamma \vdash v : \tau$ and $\Gamma, x : \tau, \Gamma' \vdash A : \widetilde{B}$, then $\Gamma, \Gamma'[v/x] \vdash A[v/x] : \widetilde{B}$.
- If $\Gamma \vdash v : \tau$ and $\Gamma, x : \tau, \Gamma' \vdash \phi$, then $\Gamma, \Gamma'[v/x] \vdash \phi[v/x]$.
- If $\Gamma \vdash v : \tau$ and $\Gamma, x : \tau, \Gamma' \vDash \phi$, then $\Gamma, \Gamma'[v/x] \vDash \phi[v/x]$.
- If $\Gamma \vdash A : \widetilde{B}$ and $\Gamma, X : \widetilde{B}, \Gamma' \vdash A' : \widetilde{B'}$, then $\Gamma, \Gamma'[A/X] \vdash A'[A/X] : \widetilde{B'}$.
- If $\Gamma \vdash A : \widetilde{B}$ and $\Gamma, X : \widetilde{B}, \Gamma' \vdash \phi$, then $\Gamma, \Gamma'[A/X] \vdash \phi[A/X]$.
- If $\Gamma \vdash A : \widetilde{B}$ and $\Gamma, X : \widetilde{B}, \Gamma' \vDash \phi$, then $\Gamma, \Gamma'[A/X] \vDash \phi[A/X]$.
- If $\vdash \Gamma_1, \Gamma_2, \Gamma_3$ and $\Gamma_1, \Gamma_2 \vdash \phi$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \phi$.
- If $\vdash \Gamma_1, \Gamma_2, \Gamma_3$ and $\Gamma_1, \Gamma_2 \vdash A : \widetilde{B}$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vdash A : \widetilde{B}$.
- If $\vdash \Gamma_1, \Gamma_2, \Gamma_3$ and $\Gamma_1, \Gamma_2 \vDash \phi$, then $\Gamma_1, \Gamma_2, \Gamma_3 \vDash \phi$.
- If $\Gamma \vdash \tau_1 <: \tau_2$, $\vdash \Gamma, x : \tau_1, \Gamma'$ and $\Gamma, x : \tau_2, \Gamma' \vdash A : \widetilde{B}$, then $\Gamma, x : \tau_1, \Gamma' \vdash A : \widetilde{B}$.

- If $\Gamma \vdash \tau_1 <: \tau_2$, $\vdash \Gamma, x : \tau_1, \Gamma'$ and $\Gamma, x : \tau_2, \Gamma' \vdash \phi$, then $\Gamma, x : \tau_1, \Gamma' \vdash \phi$.
- If $\Gamma \vdash \tau_1 <: \tau_2$ and $\Gamma, x : \tau_2, \Gamma' \vDash \phi$, then $\Gamma, x : \tau_1, \Gamma' \vDash \phi$.
- If $x \notin \mathit{fv}(\Gamma', \phi)$ and $\Gamma, x : \tau_0, \Gamma' \vdash \phi$, then $\Gamma, \Gamma' \vdash \phi$.
- If $x \notin \mathit{fv}(\Gamma', A)$ and $\Gamma, x : \tau_0, \Gamma' \vdash A : \widetilde{B}$, then $\Gamma, \Gamma' \vdash A : \widetilde{B}$.
- If $\Gamma, x : \tau, \Gamma' \vdash \phi$ and τ is not a refinement type, then $x \notin \mathit{fv}(\Gamma', \phi)$.
- If $\Gamma, x : \tau, \Gamma' \vdash A : \widetilde{B}$ and τ is not a refinement type, then $x \notin \mathit{fv}(\Gamma', A)$.
- If $\Gamma, x : \tau, \Gamma' \vDash \phi$ and τ is not a refinement type, then $x \notin \mathit{fv}(\Gamma', \phi)$ and $\Gamma, \Gamma' \vDash \phi$.
- If $\alpha \notin \mathit{fv}(\Gamma', \phi)$ and $\Gamma, \alpha, \Gamma' \vdash \phi$, then $\Gamma, \Gamma' \vdash \phi$.
- If $\alpha \notin \mathit{fv}(\Gamma', \phi)$ and $\Gamma, \alpha, \Gamma' \vDash \phi$, then $\Gamma, \Gamma' \vDash \phi$.

Assumption 34.

- $\vdash \mathit{ty}_{cps}(p)$ for all p.

Lemma 35 (Weakening)**.** *Assume that* $\vdash \Gamma_1, \Gamma_2, \Gamma_3$*.*

- *If* $\Gamma_1, \Gamma_3 \vdash \tau$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \tau$*.*
- *If* $\Gamma_1, \Gamma_3 \vdash c : \tau$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash c : \tau$*.*
- *If* $\Gamma_1, \Gamma_3 \vdash \tau_1 <: \tau_2$*, then* $\Gamma_1, \Gamma_2, \Gamma_3 \vdash \tau_1 <: \tau_2$*.*

Proof. By induction on the derivation. Assumption 33 is used. □

Lemma 36 (Narrowing)**.** *Assume that* $\Gamma \vdash \tau_1 <: \tau_2$*.*

- *If* $\vdash \Gamma, x : \tau_1, \Gamma'$ *and* $\Gamma, x : \tau_2, \Gamma' \vdash \tau$*, then* $\Gamma, x : \tau_1, \Gamma' \vdash \tau$*.*
- *If* $\vdash \Gamma, x : \tau_1, \Gamma'$ *and* $\Gamma, x : \tau_2, \Gamma' \vdash c : \tau$*, then* $\Gamma, x : \tau_1, \Gamma' \vdash c : \tau$*.*
- *If* $\Gamma, x : \tau_2, \Gamma' \vdash \tau_1 <: \tau_2$*, then* $\Gamma, x : \tau_1, \Gamma' \vdash \tau_1 <: \tau_2$*.*

Proof. By induction on the derivation. Assumption 33 is used. □

Lemma 37 (Remove unused type bindings)**.**

- *If* $x \notin \mathit{fv}(\Gamma')$ *and* $\vdash \Gamma, x : \tau_0, \Gamma'$*, then* $\vdash \Gamma, \Gamma'$*.*
- *If* $x \notin \mathit{fv}(\Gamma', \tau)$ *and* $\Gamma, x : \tau_0, \Gamma' \vdash \tau$*, then* $\Gamma, \Gamma' \vdash \tau$*.*

Proof. By induction on the derivation. The case for (WTc-Rfn) uses Assumption 33. □

Lemma 38 (Variables of non-refinement types do not apper in types)**.** *Assume that* τ_0 *is not a refinement type.*

- *If* $\vdash \Gamma, x : \tau_0, \Gamma'$*, then* $x \notin \mathit{fv}(\Gamma')$*.*
- *If* $\Gamma, x : \tau_0, \Gamma' \vdash \tau$*, then* $x \notin \mathit{fv}(\Gamma', \tau)$*.*

Proof. By induction on the derivation. The case for (WTc-Rfn) uses Assumption 33. □

Lemma 39 (Remove non-refinement type bindings)**.** *Assume that* τ_0 *is not a refinement type.*

1. *If* $\vdash \Gamma, x : \tau_0, \Gamma'$*, then* $\vdash \Gamma, \Gamma'$*.*
2. *If* $\Gamma, x : \tau_0, \Gamma' \vdash \tau$*, then* $\Gamma, \Gamma' \vdash \tau$*.*
3. *If* $x \notin \mathit{fv}(c)$ *and* $\Gamma, x : \tau_0, \Gamma' \vdash c : \tau$*, then* $\Gamma, \Gamma' \vdash c : \tau$*.*
4. *If* $\Gamma, x : \tau_0, \Gamma' \vdash \tau_1 <: \tau_2$*, then* $\Gamma, \Gamma' \vdash \tau_1 <: \tau_2$*.*

Proof.

1. Immediate by Lemma 38 and 37.
2. Immediate by Lemma 38 and 37.

3. By induction on the derivation. The case for (Tc-PApp) uses Assumption 33.

4. By induction on the derivation. The case for (Sc-Rfn) uses Assumption 33.

□

Lemma 40 (Remove unused type variable bindings)**.**

- *If* $\alpha \notin fv(\Gamma')$ *and* $\vdash \Gamma, \alpha, \Gamma'$*, then* $\vdash \Gamma, \Gamma'$*.*
- *If* $\alpha \notin fv(\Gamma', \tau)$ *and* $\Gamma, \alpha, \Gamma' \vdash \tau$*, then* $\Gamma, \Gamma' \vdash \tau$*.*
- *If* $\alpha \notin fv(\Gamma', \tau_1, \tau_2)$ *and* $\Gamma, \alpha, \Gamma' \vdash \tau_1 <: \tau_2$*, then* $\Gamma, \Gamma' \vdash \tau_1 <: \tau_2$*.*

Proof. By induction on the derivation. The case for (WTSc-Rfn) and (Sc-Rfn) uses Assumption 33. □

Lemma 41 (Substitution)**.** *Assume that* $\Gamma \vdash v : \tau_0$*.*

- *If* $\vdash \Gamma, x : \tau_0, \Gamma'$*, then* $\vdash \Gamma, \Gamma'[v/x]$*.*
- *If* $\Gamma, x : \tau_0, \Gamma' \vdash \tau$*, then* $\Gamma, \Gamma'[v/x] \vdash \tau[v/x]$*.*
- *If* $\Gamma, x : \tau_0, \Gamma' \vdash c : \tau$*, then* $\Gamma, \Gamma'[v/x] \vdash c[v/x] : \tau[v/x]$*.*
- *If* $\Gamma, x : \tau_0, \Gamma' \vdash \tau_1 <: \tau_2$*, then* $\Gamma, \Gamma'[v/x] \vdash \tau_1[v/x] <: \tau_2[v/x]$*.*

Proof. By induction on the derivation. Assumption 33 is used. □

Lemma 42 (Predicate substitution)**.** *Assume that* $\Gamma \vdash A : \widetilde{B}$*.*

- *If* $\vdash \Gamma, X : \widetilde{B}, \Gamma'$*, then* $\vdash \Gamma, \Gamma'[A/X]$*.*
- *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash \tau$*, then* $\Gamma, \Gamma'[A/X] \vdash \tau[A/X]$*.*
- *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash c : \tau$*, then* $\Gamma, \Gamma'[A/X] \vdash c[A/X] : \tau[A/X]$*.*
- *If* $\Gamma, X : \widetilde{B}, \Gamma' \vdash \tau_1 <: \tau_2$*, then* $\Gamma, \Gamma'[A/X] \vdash \tau_1[A/X] <: \tau_2[A/X]$*.*

Proof. By induction on the derivation. Assumption 33 is used. □

Lemma 43 (Type substitution)**.** *Assume that* $\Gamma \vdash \tau_0$*.*

- *If* $\vdash \Gamma, \alpha, \Gamma'$*, then* $\vdash \Gamma, \Gamma'[\tau_0/\alpha]$*.*
- *If* $\Gamma, \alpha, \Gamma' \vdash \tau$*, then* $\Gamma, \Gamma'[\tau_0/\alpha] \vdash \tau[\tau_0/\alpha]$*.*
- *If* $\Gamma, \alpha, \Gamma' \vdash c : \tau$*, then* $\Gamma, \Gamma'[\tau_0/\alpha] \vdash c[\tau_0/\alpha] : \tau[\tau_0/\alpha]$*.*
- *If* $\Gamma, \alpha, \Gamma' \vdash \tau_1 <: \tau_2$*, then* $\Gamma, \Gamma'[\tau_0/\alpha] \vdash \tau_1[\tau_0/\alpha] <: \tau_2[\tau_0/\alpha]$*.*

Proof. By induction on the derivation. Assumption 33 is used. □

Lemma 44 (Well-formedness of typing contexts from that of types)**.** *If* $\Gamma \vdash \tau$*, then* $\vdash \Gamma$*.*

Proof. By induction on the derivation. The case for (WTc-Rfn) uses Assumption 33. □

Lemma 45 (Well-formedness of types from typings)**.** *If* $\Gamma \vdash c : \tau$*, then* $\Gamma \vdash \tau$*.*

Proof. By induction on the derivation.

Case (Tc-CVar)**:** By Assumption 33.

Case (Tc-Var)**:** By Lemma 35.

Case (Tc-Prim)**:** By Assumption 34 and Lemma 35.

Case (Tc-Fun)**:** By the IH, Lemma 39, and (WTc-Fun).

Case (Tc-App)**:** By the IH, inversion, and Lemma 41.

Case (Tc-TAbs)**:** By the IH and (WTc-TPoly).

Case (Tc-TApp)**:** By the IH, inversion, and Lemma 43.

Case (Tc-PAbs)**:** By the IH and (WTc-PPoly).

Case (Tc-PApp)**:** By the IH, inversion, and Lemma 42.

Case (Tc-If)**:** By the IH and Lemma 37.

Case (Tc-Ascr) **and** (Tc-Sub)**:** Immediate.

□

Lemma 46 (Reflexivity)**.** *If* $\Gamma \vdash \tau$*, then* $\Gamma \vdash \tau <: \tau$*.*

Proof. By induction on the derivation. The case for (WTc-Rfn) uses Assumption 33. □

Lemma 47 (Transitivity)**.** *If* $\Gamma \vdash \tau_1 <: \tau_2$ *and* $\Gamma \vdash \tau_2 <: \tau_3$*, then* $\Gamma \vdash \tau_1 <: \tau_3$*.*

Proof. By induction on the structure of τ_2. Assumption 33, Lemma 36, and 35 are used. □

Lemma 48 (Inversion)**.**

- *If* $\Gamma \vdash x : \tau$*, then either*
  - $\vdash \Gamma$ *and* $\Gamma \vdash \{z : B \mid z = x\} <: \tau$ *(if* $\Gamma(x) = \{z : B \mid \phi\}$ *for some* z, B *and* ϕ*)*
  - $\vdash \Gamma$ *and* $\Gamma \vdash \Gamma(x) <: \tau$ *(otherwise)*
- *If* $\Gamma \vdash p : \tau$*, then* $\vdash \Gamma$ *and* $\Gamma \vdash ty_{cps}(p) <: \tau$*.*
- *If* $\Gamma \vdash \mathbf{rec}(f : (x : \tau_1) \to \tau_2, x : \tau_1).c : \tau$*, then* $\Gamma, f : (x : \tau_1) \to \tau_2, x : \tau_1 \vdash c : \tau_2$ *and* $\Gamma \vdash (x : \tau_1) \to \tau_2 <: \tau$*.*
- *If* $\Gamma \vdash \Lambda\alpha.c : \tau$*, then* $\Gamma, \alpha \vdash c : \tau'$ *and* $\Gamma \vdash \forall\alpha.\tau' <: \tau$ *for some* τ'*.*
- *If* $\Gamma \vdash \{(\mathsf{op}_i = v_i)_i\} : \tau$*, then* $(\Gamma \vdash v_i : \tau_i)_i$ *and* $\Gamma \vdash \{\mathsf{op}_i : \tau_i\} <: \tau$ *for some* $(\tau_i)_i$*.*
- *If* $\Gamma \vdash c\ v : \tau$*, then* $\Gamma \vdash c : (x : \tau_1) \to \tau_2$*,* $\Gamma \vdash v : \tau_1$ *and* $\Gamma \vdash \tau_2[v/x] <: \tau$ *for some* x, τ_1 *and* τ_2*.*
- *If* $\Gamma \vdash c\ \widetilde{A} : \tau$*, then* $\Gamma \vdash c : \forall\widetilde{X : \widetilde{B}}.\tau'$*,* $\Gamma \vdash \widetilde{A : \widetilde{B}}$ *and* $\Gamma \vdash \tau'[\widetilde{A/X}] <: \tau$ *for some* $\widetilde{X}, \widetilde{\widetilde{B}}$ *and* τ'*.*
- *If* $\Gamma \vdash c\ \tau' : \tau$*, then* $\Gamma \vdash c : \forall\alpha.\tau_1$*,* $\Gamma \vdash \tau'$ *and* $\Gamma \vdash \tau_1[\tau'/\alpha] <: \tau$ *for some* α *and* τ_1*.*
- *If* $\Gamma \vdash v\#\mathsf{op} : \tau$*, then* $\Gamma \vdash v : \{\ldots, \mathsf{op} : \tau, \ldots\}$*.*
- *If* $\Gamma \vdash (c : \tau') : \tau$*, then* $\Gamma \vdash c : \tau'$ *and* $\Gamma \vdash \tau' <: \tau$*.*
- *If* $\Gamma \vdash \mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2 : \tau$*, then* $\Gamma \vdash v : \{z : \mathrm{bool} \mid \phi\}$*,* $\Gamma, v = \mathbf{true} \vdash c_1 : \tau'$*,* $\Gamma, v = \mathbf{false} \vdash c_2 : \tau'$*, and* $\Gamma \vdash \tau' <: \tau$ *for some* z, ϕ *and* τ'*.*

Proof. By induction on the derivation. Lemma 46 and 47 are used. □

Lemma 49 (Inversion for CPS-transformed computations)**.** *If* $\Gamma \vdash \Lambda\alpha.\lambda h : \tau_h.\lambda k : \tau_k.c : \tau$ *and neither* τ_h *nor* τ_k *is a refinement type, then there exists some* τ' *such that*

- $\Gamma, \alpha, h : \tau_h, k : \tau_k \vdash c : \tau'$ *and*
- $\Gamma \vdash \forall\alpha.\tau_h \to \tau_k \to \tau' <: \tau$ *.*

Proof. By Lemma 48, (Sc-Poly), (Sc-Fun), and Lemma 47. □

Lemma 50 (Inversion for the specific form of application)**.** *If* $\Gamma \vdash c\ \tau_0\ v_1\ v_2 : \tau$*, then there exist some* τ', τ_1*, and* τ_2 *such that*

- $\Gamma \vdash c : \tau'$*,*
- $\Gamma \vdash v_1 : \tau_1$*, and*
- $\Gamma \vdash v_2 : \tau_2$ *.*

In addition, if $\Gamma \vdash \tau_1' <: \tau_1$ *and* $\Gamma \vdash \tau_2' <: \tau_2$ *for some* τ_1' *and* τ_2' *and neither* τ_1' *nor* τ_2' *is a refinement type, then* $\Gamma \vdash \tau' <: \forall\alpha.\tau_1' \to \tau_2' \to \tau$ *where* α *is fresh.*

Proof. The first half is by Lemma 48. The second half is by Lemma 45, 35 and 47 with the results of the first half. □

7.2 Forward type preservation

Assumption 51.

- If $\Gamma \vdash \phi$, then $[\![\Gamma]\!] \vdash \phi$.
- If $\Gamma \vdash A : \widetilde{B}$, then $[\![\Gamma]\!] \vdash A : \widetilde{B}$.
- If $\Gamma \models \phi$, then $[\![\Gamma]\!] \models \phi$.

Assumption 52.

- $[\![ty(p)]\!] = ty_{cps}([\![p]\!])$.
- If $ty(p) = \{x : B \mid \phi\}$ for some x, B and ϕ, then $[\![p]\!] = p$.

Lemma 53 (CPS transformation preserves free variables in types)**.**

- $fv([\![T]\!]) = fv(T)$.
- $fv([\![C]\!]) = fv(C)$.
- $fv([\![\Sigma]\!]) = fv(\Sigma)$.

Proof. By simultaneous induction on the structure of types. □

Lemma 54 (CPS transformation is homomorphic for substitution)**.**

- $[\![T[v/x]]\!] = [\![T]\!][[\![v]\!]/x]$.
- $[\![C[v/x]]\!] = [\![C]\!][[\![v]\!]/x]$.
- $[\![\Sigma[v/x]]\!] = [\![\Sigma]\!][[\![v]\!]/x]$.
- $[\![T[A/X]]\!] = [\![T]\!][A/X]$.
- $[\![C[A/X]]\!] = [\![C]\!][A/X]$.
- $[\![\Sigma[A/X]]\!] = [\![\Sigma]\!][A/X]$.

Proof. By simultaneous induction on the structure of types. The case for $T = \{x : B \mid \phi\}$ uses Assumption 52. □

Lemma 55 (CPS transformation preserves well-formedness)**.**

- *If* $\vdash \Gamma$*, then* $\vdash [\![\Gamma]\!]$.
- *If* $\Gamma \vdash T$*, then* $[\![\Gamma]\!] \vdash [\![T]\!]$.
- *If* $\Gamma \vdash C$*, then* $[\![\Gamma]\!] \vdash [\![C]\!]$.
- *If* $\Gamma \vdash \Sigma$*, then* $[\![\Gamma]\!] \vdash [\![\Sigma]\!]$.

Proof. By simultaneous induction on the derivations. Lemma 35 is used. The case for (WT-Rfn) uses Assumption 51. □

Lemma 56 (CPS transformation preserves subtyping)**.**

- *If* $\Gamma \vdash T_1 <: T_2$*, then* $[\![\Gamma]\!] \vdash [\![T_1]\!] <: [\![T_2]\!]$.
- *If* $\Gamma \vdash C_1 <: C_2$*, then* $[\![\Gamma]\!] \vdash [\![C_1]\!] <: [\![C_2]\!]$.
- *If* $\Gamma \vdash \Sigma_1 <: \Sigma_2$*, then* $[\![\Gamma]\!] \vdash [\![\Sigma_1]\!] <: [\![\Sigma_2]\!]$.

Proof. By simultaneous induction on the derivations. Lemma 35 is used. The case for (S-Rfn) uses Assumption 51. □

Theorem 57 (Forward type preservation)**.**

1. *If* $\Gamma \vdash v : T$*, then* $[\![\Gamma]\!] \vdash [\![v]\!] : [\![T]\!]$.
2. *If* $\Gamma \vdash c : C$*, then* $[\![\Gamma]\!] \vdash [\![c]\!] : [\![C]\!]$.

Proof. By simultaneous induction on the typing derivation of the source language.

1. **Case** (T-CVar)**:** By Lemma 55, definition of CPS transformation of typing contexts, and (Tc-CVar).
   Case (T-Var)**:** By Lemma 55, definition of CPS transformation of typing contexts, and (Tc-Var).
   Case (T-Prim)**:** By Lemma 55, Assumption 52, and (Tc-Prim).
   Case (T-Fun)**:** By the IH and (Tc-Fun).
   Case (T-VSub)**:** By the IH, Lemma 56, Lemma 55 and (Tc-Sub).
2. **Case** (T-Ret)**:** we have
   - $c = \textbf{return}\ v$,
   - $C = \{\} \triangleright T \ / \ \square$, and
   - $\Gamma \vdash v : T$

   for some v and T. Then, we have
   - $[\![c]\!] = \Lambda\alpha.\lambda h : \{\}.\lambda k : [\![T]\!] \to \alpha.k\ [\![v]\!]$ and
   - $[\![C]\!] = \forall\alpha.\{\} \to ([\![T]\!] \to \alpha) \to \alpha$.

   By the IH, we have
   - $[\![\Gamma]\!] \vdash [\![v]\!] : [\![T]\!]$.

   We have the conclusion by the following derivation with Lemma 35:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_{\alpha,h,k} \vdash k : [\![T]\!] \to \alpha}\,(\textsc{Tc-Var}) \qquad \Gamma_{\alpha,h,k} \vdash [\![v]\!] : [\![T]\!]}{[\![\Gamma]\!], \alpha, h : \{\}, k : [\![T]\!] \to \alpha \vdash k\ [\![v]\!] : \alpha}\,(\textsc{Tc-App})}{[\![\Gamma]\!], \alpha, h : \{\} \vdash \lambda k[\![T]\!] \to \alpha.k\ [\![v]\!] : ([\![T]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-Lam})}{[\![\Gamma]\!], \alpha \vdash \lambda h : \{\}.\lambda k : [\![T]\!] \to \alpha.k\ [\![v]\!] : \{\} \to ([\![T]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-Lam})}{[\![\Gamma]\!] \vdash \Lambda\alpha.\lambda h : \{\}.\lambda k : [\![T]\!] \to \alpha.k\ [\![v]\!] : \forall\alpha.\{\} \to ([\![T]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-TAbs})$$

   where $\Gamma_{\alpha,h,k} \stackrel{\text{def}}{=} [\![\Gamma]\!], \alpha, h : \{\}, k : [\![T]\!] \to \alpha$.

   Case (T-App)**:** By the IH, Lemma 54 and (Tc-App).
   Case (T-If)**:** By the IH, (Tc-If) and (Tc-Ascr).
   Case (T-CSub)**:** similar to the case for (T-VSub).
   Case (T-LetP)**:** We have
   - $c = \textbf{let}\ x = c_1\ \textbf{in}\ c_2$,
   - $C = \Sigma \triangleright T_2 \ / \ \square$,
   - $\Gamma \vdash c_1 : \Sigma \triangleright T_1 \ / \ \square$,
   - $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T_2 \ / \ \square$, and
   - $x \notin \mathit{fv}(T_2) \cup \mathit{fv}(\Sigma)$

   for some x, c_1, c_2, Σ, T_1 and T_2. Then we have
   - $[\![c]\!] = \Lambda\alpha.\lambda h : [\![\Sigma]\!].\lambda k : [\![T_2]\!] \to \alpha.[\![c_1]\!]\ \alpha\ h\ (\lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k)$ and
   - $[\![C]\!] = \forall\alpha.[\![\Sigma]\!] \to ([\![T_2]\!] \to \alpha) \to \alpha$.

   By Lemma 53, we have
   - $x \notin \mathit{fv}([\![T_2]\!]) \cup \mathit{fv}([\![\Sigma]\!])$.

   Also, by the IHs, we have
   - $[\![\Gamma]\!] \vdash [\![c_1]\!] : \forall\beta.[\![\Sigma]\!] \to ([\![T_1]\!] \to \beta) \to \beta$ and
   - $[\![\Gamma]\!], x : [\![T_1]\!] \vdash [\![c_2]\!] : \forall\gamma.[\![\Sigma]\!] \to ([\![T_2]\!] \to \gamma) \to \gamma$.

   We have the conclusion by the following derivations with Lemma 35:

$$(A) : \quad \dfrac{\dfrac{\dfrac{\Gamma_{\alpha,h,k} \vdash [\![c_1]\!] : \forall\beta.[\![\Sigma]\!] \to ([\![T_1]\!] \to \beta) \to \beta \qquad \Gamma_{\alpha,h,k} \vdash \alpha}{\Gamma_{\alpha,h,k} \vdash [\![c_1]\!]\ \alpha : [\![\Sigma]\!] \to ([\![T_1]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-TApp}) \qquad \dfrac{}{\Gamma_{\alpha,h,k} \vdash h : [\![\Sigma]\!]}\,(\textsc{Tc-Var})}{\Gamma_{\alpha,h,k} \vdash [\![c_1]\!]\ \alpha\ h : ([\![T_1]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-App})}{[\![\Gamma]\!]_{\alpha,h,k} \vdash [\![c_1]\!]\ \alpha\ h : ([\![T_1]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-App})$$

$$(B) : \quad \dfrac{\dfrac{\Gamma_{\alpha,h,k,x} \vdash [\![c_2]\!] : \forall\gamma.[\![\Sigma]\!] \to ([\![T_2]\!] \to \gamma) \to \gamma \qquad \Gamma_{\alpha,h,k,x} \vdash \alpha}{\Gamma_{\alpha,h,k,x} \vdash [\![c_2]\!]\ \alpha : [\![\Sigma]\!] \to ([\![T_2]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-TApp}) \qquad \dfrac{}{\Gamma_{\alpha,h,k,x} \vdash h : [\![\Sigma]\!]}\,(\textsc{Tc-Var})}{\Gamma_{\alpha,h,k,x} \vdash [\![c_2]\!]\ \alpha\ h : ([\![T_2]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-App})$$

$$\dfrac{\dfrac{(A)\quad \dfrac{\dfrac{(B)\quad \overline{\Gamma_{\alpha,h,k,x} \vdash k : [\![T_2]\!] \to \alpha}\,(\textsc{Tc-Var})}{\Gamma_{\alpha,h,k}, x : [\![T_1]\!] \vdash [\![c_2]\!]\ \alpha\ h\ k : \alpha}\,(\textsc{Tc-App})}{\Gamma_{\alpha,h,k} \vdash \lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k : [\![T_1]\!] \to \alpha}\,(\textsc{Tc-Fun})}{[\![\Gamma]\!], \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha \vdash [\![c_1]\!]\ \alpha\ h\ (\lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k) : \alpha}\,(\textsc{Tc-App})$$

$$\dfrac{\dfrac{\dfrac{\cdots}{[\![\Gamma]\!], \alpha, h : [\![\Sigma]\!] \vdash \lambda k : [\![T_2]\!] \to \alpha.[\![c_1]\!]\ \alpha\ h\ (\lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k) : ([\![T_2]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-Fun})}{[\![\Gamma]\!], \alpha \vdash \lambda h : [\![\Sigma]\!].\lambda k : [\![T_2]\!] \to \alpha.[\![c_1]\!]\ \alpha\ h\ (\lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k) : [\![\Sigma]\!] \to ([\![T_2]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-Fun})}{[\![\Gamma]\!] \vdash \Lambda\alpha.\lambda h : [\![\Sigma]\!].\lambda k : [\![T_2]\!] \to \alpha.[\![c_1]\!]\ \alpha\ h\ (\lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k) : \forall\alpha.[\![\Sigma]\!] \to ([\![T_2]\!] \to \alpha) \to \alpha}\,(\textsc{Tc-TAbs})$$

where $\Gamma_{\alpha,h,k} \stackrel{\text{def}}{=} [\![\Gamma]\!], \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha$ and $\Gamma_{\alpha,h,k,x} \stackrel{\text{def}}{=} \Gamma_{\alpha,h,k}, x : [\![T_1]\!]$.

Case (T-LetIp)**:** We have

- $c = \mathbf{let}\ x = c_1\ \mathbf{in}\ c_2$,
- $C = \Sigma \triangleright T_2\ /\ (\forall z.C_0) \Rightarrow C_2$,
- $\Gamma \vdash c_1 : \Sigma \triangleright T_1\ /\ (\forall x.C_1) \Rightarrow C_2$,
- $\Gamma, x : T_1 \vdash c_2 : \Sigma \triangleright T_2\ /\ (\forall z.C_0) \Rightarrow C_1$, and
- $x \notin fv(T_2) \cup fv(\Sigma) \cup (fv(C_0) \setminus \{z\})$

for some $x, z, c_1, c_2, \Sigma, T_1, T_2, C_0, C_1$ and C_2. Then we have

- $[\![c]\!] = \Lambda\alpha.\lambda h : [\![\Sigma]\!].\lambda k : (z : [\![T_2]\!]) \to [\![C_0]\!].[\![c_1]\!]\ [\![C_2]\!]\ h\ (\lambda x : [\![T_1]\!].[\![c_2]\!]\ [\![C_1]\!]\ h\ k)$ and
- $[\![C]\!] = \forall\alpha.[\![\Sigma]\!] \to ((z : [\![T_2]\!]) \to [\![C_0]\!]) \to [\![C_2]\!]$.

By Lemma 53, we have

- $x \notin fv([\![T_2]\!]) \cup fv([\![\Sigma]\!]) \cup (fv([\![C_0]\!]) \setminus \{z\})$.

Also, by the IHs, we have

- $[\![\Gamma]\!] \vdash [\![c_1]\!] : \forall\alpha.[\![\Sigma]\!] \to ((x : [\![T_1]\!]) \to [\![C_1]\!]) \to [\![C_2]\!]$ and
- $[\![\Gamma]\!], x : [\![T_1]\!] \vdash [\![c_2]\!] : \forall\alpha.[\![\Sigma]\!] \to ((z : [\![T_2]\!]) \to [\![C_0]\!]) \to [\![C_1]\!]$.

We have the conclusion by a straightforward derivation like the case for (T-LetP) using those judgements shown so far and Lemma 35.

Case (T-Op)**:** (In this case, we use Lemma 54 frequently and implicitly.)
We have

- $c = \mathsf{op}\ v$,
- $C = \Sigma \triangleright T_2[\widetilde{A/X}][v/x]\ /\ (\forall y.C_1[\widetilde{A/X}][v/x]) \Rightarrow C_2[\widetilde{A/X}][v/x]$,
- $\Sigma \ni \mathsf{op} : \forall \widetilde{X : B}.(x : T_1) \to ((y : T_2) \to C_1) \to C_2$,
- $\Gamma \vdash \Sigma$,
- $\Gamma \vdash \widetilde{A : B}$, and
- $\Gamma \vdash v : T_1[\widetilde{A/X}]$

for some $x, y, v, \widetilde{X}, \widetilde{A}, \widetilde{\widetilde{B}}, \Sigma, T_1, T_2, C_1$ and C_2. Then, we have

- $[\![c]\!] = \Lambda\alpha.\lambda h : [\![\Sigma]\!].\lambda k : (y : [\![T_2]\!][\widetilde{A/X}][[\![v]\!]/x] \to [\![C_1]\!][\widetilde{A/X}][[\![v]\!]/x]).$
  $h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y' : [\![T_2]\!][\widetilde{A/X}][[\![v]\!]/x].k\ y')$,
- $[\![C]\!] = \forall\alpha.[\![\Sigma]\!] \to ((y : [\![T_2]\!][\widetilde{A/X}][[\![v]\!]/x]) \to [\![C_1]\!][\widetilde{A/X}][[\![v]\!]/x]) \to [\![C_2]\!][\widetilde{A/X}][[\![v]\!]/x]$, and
- $[\![\Sigma]\!] \ni \mathsf{op} : \forall \widetilde{X : B}.(x : [\![T_1]\!]) \to ((y : [\![T_2]\!]) \to [\![C_1]\!]) \to [\![C_2]\!]$.

Also, by the IHs, we have

- $[\![\Gamma]\!] \vdash [\![v]\!] : [\![T_1]\!][\widetilde{A/X}]$.

By Assumption 51, we have

- $[\![\Gamma]\!] \vdash \widetilde{A : B}$.

We have the conclusion by a straightforward derivation like the cases for (T-Ret) and (T-LetP) using those judgements shown so far and Lemma 35.

Case (T-Hndl)**:** We have

- $c = \mathbf{with}\ h\ \mathbf{handle}\ c_0$,
- $h = \{\mathbf{return}\ x_r \mapsto c_r, (\mathsf{op}_i(x_i, k_i) \mapsto c_i)_i\}$,
- $\Gamma \vdash c_0 : \Sigma_0 \triangleright T_r\ /\ (\forall x_r.C_1) \Rightarrow C$,

- $\Gamma, x_r : T_r \vdash c_r : C_1$,
- $\left(\Gamma, \widetilde{X_i : \widetilde{B}_i}, x_i : T_{i1}, k_i : (y_i : T_{i2}) \to C_{i1} \vdash c_i : C_{i2}\right)_i$, and
- $\Sigma_0 = \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.(x_i : T_{i1}) \to ((y_i : T_{i2}) \to C_{i1}) \to C_{2i})_i\}$

Then, we have

- $[\![c]\!] = [\![c_0]\!]\ [\![C]\!]\ [\![h^{ops}]\!]\ [\![h^{ret}]\!]$,
- $[\![h^{ret}]\!] = \lambda x_r : [\![T_r]\!].[\![c_r]\!]$,
- $[\![h^{ops}]\!] = \{(\mathsf{op}_i = \Lambda \widetilde{X_i : \widetilde{B}_i}.\lambda x_i : [\![T_{i1}]\!].\lambda k_i : (y_i : [\![T_{i2}]\!]) \to [\![C_{i1}]\!].[\![c_i]\!])_i\}$, and
- $[\![\Sigma_0]\!] = \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B}_i}.(x_i : [\![T_{i1}]\!]) \to ((y_i : [\![T_{i2}]\!]) \to [\![C_{i1}]\!]) \to [\![C_{i2}]\!])_i\}$.

Also, by the IHs, we have

- $[\![\Gamma]\!] \vdash [\![c_0]\!] : \forall \alpha.[\![\Sigma_0]\!] \to ((x_r : [\![T_r]\!]) \to [\![C_1]\!]) \to [\![C]\!]$,
- $[\![\Gamma]\!], x_r : [\![T_r]\!] \vdash [\![c_r]\!] : [\![C_1]\!]$, and
- $\left([\![\Gamma]\!], \widetilde{X_i : \widetilde{B}_i}, x_i : [\![T_{i1}]\!], k_i : (y_i : [\![T_{i2}]\!]) \to [\![C_{i1}]\!] \vdash [\![c_i]\!] : [\![C_{i2}]\!]\right)_i$.

We have the conclusion by a straightforward derivation like the cases for (T-Ret) and (T-LetP) using those judgements shown so far and Lemma 35.

□

7.3 Backward type preservation

For the backward direction, we define some notations.

Definition 58. Γ is *cps-wellformed* if for all $(x : \tau) \in \Gamma$, it holds that $\tau = [\![T]\!]$ for some T.

Definition 59. *rmtv* is a function which removes all bindings of type variables from a typing context. Formally, it is defined as follows:

$$rmtv(\emptyset) \stackrel{\text{def}}{=} \emptyset \qquad rmtv(\Gamma, x : \tau) \stackrel{\text{def}}{=} rmtv(\Gamma), x : \tau$$
$$rmtv(\Gamma, X : \widetilde{B}) \stackrel{\text{def}}{=} rmtv(\Gamma), X : \widetilde{B} \qquad rmtv(\Gamma, \alpha) \stackrel{\text{def}}{=} rmtv(\Gamma)$$

Lemma 60 (CPS-wellformed target typing contexts have corresponding source ones)**.** *If Γ is cps-wellformed, then there exists some Γ' such that $[\![\Gamma']\!] = rmtv(\Gamma)$.*

Proof. By induction on the structure of Γ. □

Since the CPS transformation is injective, there is only one Γ' which satisfies the equation in Lemma 60. Therefore, we define a function $(\!|-|\!)$ that maps Γ to Γ':

Definition 61. Let Γ be a cps-wellformed typing context in the target language. We define $(\!|\Gamma|\!)$ to be the typing context in the source language such that $[\![(\!|\Gamma|\!)]\!] = rmtv(\Gamma)$.

Assumption 62. Assume that Γ is cps-wellformed.

- If $\Gamma \vdash \phi$, then $(\!|\Gamma|\!) \vdash \phi$.
- If $\Gamma \vdash A : \widetilde{B}$, then $(\!|\Gamma|\!) \vdash A : \widetilde{B}$.
- If $\Gamma \models \phi$, then $(\!|\Gamma|\!) \models \phi$.

Lemma 63 (Computation types in the specific form of subtyping are pure)**.** *If $\Gamma \vdash [\![C]\!] <: \forall \alpha.\tau_1 \to (\tau_2 \to \beta) \to \tau_4$ and $\beta \in \Gamma$, then $C = \Sigma \triangleright T \ /\ \square$ (for some Σ and T), and $\tau_4 = \beta$.*

Proof. Assume that $C = \Sigma \triangleright T\ /\ (\forall x.C_1) \Rightarrow C_2$ for some Σ, T, x, C_1 and C_2. Then, we have

$$\Gamma \vdash \forall \gamma.[\![\Sigma]\!] \to ((x : [\![T]\!]) \to [\![C_1]\!]) \to [\![C_2]\!] <: \forall \alpha.\tau_1 \to (\tau_2 \to \beta) \to \tau_4$$

where γ is fresh. By inversion, we have $\Gamma, \alpha, h : \tau_1, x : [\![T]\!] \vdash \beta <: [\![C_1]\!]$, that is, $\Gamma, \alpha, h : \tau_1, x : [\![T]\!] \vdash \beta <: \forall \delta.\tau_5$ for some τ_5 and δ. This is contradictory since there is no subtyping rule for such a judgment.

Therefore, $C = \Sigma \triangleright T\ /\ \square$ for some Σ and T. In this case, we have

$$\Gamma \vdash \forall \gamma.[\![\Sigma]\!] \to ([\![T]\!] \to \gamma) \to \gamma <: \forall \alpha.\tau_1 \to (\tau_2 \to \beta) \to \tau_4$$

where γ is fresh. By inversion, we have

- $\Gamma, \alpha \vdash \tau_6$,
- $\Gamma, \alpha, h : \tau_1, x : [\![T]\!][\tau_6/\gamma] \vdash \beta <: \gamma[\tau_6/\gamma]$, and
- $\Gamma, \alpha, h : \tau_1, x : [\![T]\!][\tau_6/\gamma] \vdash \gamma[\tau_6/\gamma] <: \tau_4$

for some τ_6. The second judgment can be derived by only (Sc-TVar) where $\gamma[\tau_6/\gamma] = \beta$. Therefore, the third judgment becomes $\Gamma, \alpha, h : \tau_1, x : [\![T]\!][\tau_6/\gamma] \vdash \beta <: \tau_4$, which can be derived by only (Sc-TVar) where $\tau_4 = \beta$. □

Lemma 64 (Computation types can be assumed to be impure)**.** *If $\Gamma \vdash c : C$, then w.l.o.g., we can assume that $C = \Sigma \triangleright T \;/\; (\forall x.C_1) \Rightarrow C_2$ for some Σ, T, x, C_1 and C_2.*

Proof.

Case $C = \Sigma \triangleright T \;/\; (\forall x.C_1) \Rightarrow C_2$**:** Immediate.

Case $C = \Sigma \triangleright T \;/\; \square$**:** It holds that $\Gamma \vdash \Sigma \triangleright T \;/\; \square <: \Sigma \triangleright T \;/\; (\forall x.C_0) \Rightarrow C_0$ for any C_0 such that $\Gamma \vdash C_0$. Therefore, by subsumption we have $\Gamma \vdash c : \Sigma \triangleright T \;/\; (\forall x.C_0) \Rightarrow C_0$.

□

Lemma 65 (Backward preservation on well-formedness)**.** *Assume that Γ is cps-wellformed.*

1. *If $\vdash \Gamma$, then $\vdash (\!|\Gamma|\!)$.*
2. *If $\Gamma \vdash [\![T]\!]$, then $(\!|\Gamma|\!) \vdash T$.*
3. *If $\Gamma \vdash [\![C]\!]$, then $(\!|\Gamma|\!) \vdash C$.*
4. *If $\Gamma \vdash [\![\Sigma]\!]$, then $(\!|\Gamma|\!) \vdash \Sigma$.*

Proof. By simultaneous induction on the derivation.

1. **Case** (WEc-Empty)**:** Obvious since $(\!|\emptyset|\!) = \emptyset$.

   Case (WEc-Var)**:** We have
   - $\Gamma = \Gamma', x : \tau$,
   - $\vdash \Gamma'$,
   - $x \notin dom(\Gamma')$, and
   - $\Gamma' \vdash \tau$

   for some Γ', x, and τ. By the IH, we have $\vdash (\!|\Gamma'|\!)$. Also, we have $x \notin dom((\!|\Gamma'|\!))$ since $dom(\Gamma') \supseteq dom((\!|\Gamma'|\!))$. Moreover, since Γ is cps-wellformed, $\tau = [\![T]\!]$ for some T. Then, by the IH, we have $(\!|\Gamma'|\!) \vdash T$. We have the conclusion by (WE-Var). (Note that $(\!|\Gamma|\!) = (\!|\Gamma', x : [\![T]\!]|\!) = (\!|\Gamma'|\!), x : T$.)

   Case (WEc-BVar)**:** By the IH and (WE-BVar).

   Case (WEc-PVar)**:** By the IH and (WE-PVar).

   Case (WEc-TVar)**:** By the IH. Note that $(\!|\Gamma', \alpha|\!) = (\!|\Gamma'|\!)$.

2. Case analysis on T.

   Case $T = \{z : B \mid \phi\}$**:** By Assumption 62 and (WT-Rfn).

   Case $T = (x : T_1) \to C_1$**:** By the IHs and (WT-Fun).

3. Case analysis on C.

   Case $C = \Sigma \triangleright T \;/\; \square$**:** We have $[\![C]\!] = \forall \alpha. [\![\Sigma]\!] \to ([\![T]\!] \to \alpha) \to \alpha$ for some α. By inversion, we have
   - $\Gamma, \alpha \vdash [\![\Sigma]\!]$ and
   - $\Gamma, \alpha, h : [\![\Sigma]\!] \vdash [\![T]\!]$.

   By 9, we have
   - $\Gamma, \alpha \vdash [\![T]\!]$.

   By the IHs, we have
   - $(\!|\Gamma|\!) \vdash \Sigma$ and
   - $(\!|\Gamma|\!) \vdash T$.

   Also, by (WT-Pure), we have $(\!|\Gamma|\!) \mid T \vdash \square$. Then we have the conclusion by (WT-Comp).

Case $C = \Sigma \triangleright T \;/\; (\forall x.C_1) \Rightarrow C_2$**:** We have $[\![C]\!] = \forall\alpha.[\![\Sigma]\!] \rightarrow ((x : [\![T]\!]) \rightarrow [\![C_1]\!]) \rightarrow [\![C_2]\!]$. By inversion, we have

- $\Gamma, \alpha \vdash [\![\Sigma]\!]$,
- $\Gamma, \alpha, h : [\![\Sigma]\!] \vdash [\![T]\!]$,
- $\Gamma, \alpha, h : [\![\Sigma]\!], x : [\![T]\!] \vdash [\![C_1]\!]$, and
- $\Gamma, \alpha, h : [\![\Sigma]\!], k : (x : [\![T]\!]) \rightarrow [\![C_1]\!] \vdash [\![C_2]\!]$.

By 9, we have

- $\Gamma, \alpha \vdash [\![T]\!]$,
- $\Gamma, \alpha, x : [\![T]\!] \vdash [\![C_1]\!]$, and
- $\Gamma, \alpha \vdash [\![C_2]\!]$.

By the IHs, we have

- $(\!|\Gamma|\!) \vdash \Sigma$,
- $(\!|\Gamma|\!) \vdash T$,
- $(\!|\Gamma|\!), x : T \vdash C_1$, and
- $(\!|\Gamma|\!) \vdash C_2$.

Then we have the conclusion by (WT-ATM) and (WT-Comp).

4. By the IHs and (WT-Sig).

□

Lemma 66 (Backward preservation on subtyping)**.** *Assume that* Γ *is cps-wellformed.*

1. If $\Gamma \vdash [\![T_1]\!] <: [\![T_2]\!]$*, then* $(\!|\Gamma|\!) \vdash T_1 <: T_2$.

2. If $\Gamma \vdash [\![C_1]\!] <: [\![C_2]\!]$*, then* $(\!|\Gamma|\!) \vdash C_1 <: C_2$.

3. If $\Gamma \vdash [\![\Sigma_1]\!] <: [\![\Sigma_2]\!]$*, then* $(\!|\Gamma|\!) \vdash \Sigma_1 <: \Sigma_2$.

Proof. By simultaneous induction on the derivation.

1. Case analysis on T_1 and T_2.

Case $T_1 = \{z : B \mid \phi_1\}$ **and** $T_2 = \{z : B \mid \phi_2\}$**:** We have

- $[\![T_1]\!] = \{z : B \mid \phi_1\}$ and
- $[\![T_2]\!] = \{z : B \mid \phi_2\}$.

We have the conclusion by Assumption 62 and (S-Rfn).

Case $T_1 = (x : T_{10}) \rightarrow C_1$ **and** $T_1 = (x : T_{10}) \rightarrow C_1$**:** We have

- $[\![T_1]\!] = (x : [\![T_{10}]\!]) \rightarrow [\![C_1]\!]$ and
- $[\![T_2]\!] = (x : [\![T_{20}]\!]) \rightarrow [\![C_2]\!]$.

We have the conclusion by the IHs and (S-Fun).

Otherwise: Contradictory since there is no applicable rule.

2. Case analysis on C_1 and C_2.

Case $C_1 = \Sigma_1 \triangleright T_1 \;/\; \square$ **and** $C_2 = \Sigma_2 \triangleright T_2 \;/\; \square$**:** We have

- $[\![C_1]\!] = \forall\alpha.[\![\Sigma_1]\!] \rightarrow ([\![T_1]\!] \rightarrow \alpha) \rightarrow \alpha$ and
- $[\![C_2]\!] = \forall\beta.[\![\Sigma_2]\!] \rightarrow ([\![T_2]\!] \rightarrow \beta) \rightarrow \beta$

for some α and β. By inversion, we have

- $\Gamma, \beta \vdash \tau$,
- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!][\tau/\alpha]$ and
- $\Gamma, \beta, h : [\![\Sigma_2]\!] \vdash [\![T_1]\!] <: [\![T_2]\!][\tau/\alpha]$

for some τ. Since CPS-transformed types do not contain type variables, we have

- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!]$ and
- $\Gamma, \beta, h : [\![\Sigma_2]\!] \vdash [\![T_1]\!] <: [\![T_2]\!]$.

By 9, we have

- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!]$ and

- $\Gamma, \beta \vdash [\![T_1]\!] <: [\![T_2]\!]$.

By the IHs, we have

- $(\!|\Gamma|\!) \vdash \Sigma_2 <: \Sigma_1$ and
- $(\!|\Gamma|\!) \vdash T_1 <: T_2$.

Also, by (S-Pure), we have $(\!|\Gamma|\!) \mid T_1 \vdash \square <: \square$. Then we have the conclusion by (S-Comp).

Case $C_1 = \Sigma_1 \triangleright T_1 \ / \ (\forall x.C_{11}) \Rightarrow C_{12}$ **and** $C_2 = \Sigma_2 \triangleright T_2 \ / \ (\forall x.C_{21}) \Rightarrow C_{22}$**:** We have

- $[\![C_1]\!] = \forall \alpha.[\![\Sigma_1]\!] \rightarrow ((x : [\![T_1]\!]) \rightarrow [\![C_{11}]\!]) \rightarrow [\![C_{12}]\!]$ and
- $[\![C_2]\!] = \forall \beta.[\![\Sigma_2]\!] \rightarrow ((x : [\![T_2]\!]) \rightarrow [\![C_{21}]\!]) \rightarrow [\![C_{22}]\!]$.

By inversion, we have

- $\Gamma, \beta \vdash \tau$,
- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!][\tau/\alpha]$,
- $\Gamma, \beta, h : [\![\Sigma_2]\!] \vdash [\![T_1]\!][\tau/\alpha] <: [\![T_2]\!]$,
- $\Gamma, \beta, h : [\![\Sigma_2]\!], x : [\![T_1]\!][\tau/\alpha] \vdash [\![C_{21}]\!] <: [\![C_{11}]\!][\tau/\alpha]$, and
- $\Gamma, \beta, h : [\![\Sigma_2]\!], k : (x : [\![T_2]\!]) \rightarrow [\![C_{21}]\!] \vdash [\![C_{12}]\!][\tau/\alpha] <: [\![C_2 2]\!]$.

for some τ. Since CPS-transformed types do not contain type variables, we have

- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!]$,
- $\Gamma, \beta, h : [\![\Sigma_2]\!] \vdash [\![T_1]\!] <: [\![T_2]\!]$,
- $\Gamma, \beta, h : [\![\Sigma_2]\!], x : [\![T_1]\!] \vdash [\![C_{21}]\!] <: [\![C_{11}]\!]$, and
- $\Gamma, \beta, h : [\![\Sigma_2]\!], k : (x : [\![T_2]\!]) \rightarrow [\![C_{21}]\!] \vdash [\![C_{12}]\!] <: [\![C_2 2]\!]$.

By 9, we have

- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!]$,
- $\Gamma, \beta \vdash [\![T_1]\!] <: [\![T_2]\!]$,
- $\Gamma, \beta, x : [\![T_1]\!] \vdash [\![C_{21}]\!] <: [\![C_{11}]\!]$, and
- $\Gamma, \beta \vdash [\![C_{12}]\!] <: [\![C_2 2]\!]$.

By the IHs, we have

- $(\!|\Gamma|\!) \vdash \Sigma_2 <: \Sigma_1$,
- $(\!|\Gamma|\!) \vdash T_1 <: T_2$,
- $(\!|\Gamma|\!), x : T_1 \vdash C_{21} <: C_{11}$, and
- $(\!|\Gamma|\!) \vdash C_{12} <: C_{22}$.

Then we have the conclusion by (S-ATM) and (S-Comp).

Case $C_1 = \Sigma_1 \triangleright T_1 \ / \ \square$ **and** $C_2 = \Sigma_2 \triangleright T_2 \ / \ (\forall x.C_{21}) \Rightarrow C_{22}$**:** We have

- $[\![C_1]\!] = \forall \alpha.[\![\Sigma_1]\!] \rightarrow ([\![T_1]\!] \rightarrow \alpha) \rightarrow \alpha$ and
- $[\![C_2]\!] = \forall \beta.[\![\Sigma_2]\!] \rightarrow ((x : [\![T_2]\!]) \rightarrow [\![C_{21}]\!]) \rightarrow [\![C_{22}]\!]$.

W.l.o.g., we can assume that $x \notin [\![C_{22}]\!]$. By inversion, we have

- $\Gamma, \beta \vdash \tau$,
- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!][\tau/\alpha]$,
- $\Gamma, \beta, h : [\![\Sigma_2]\!] \vdash [\![T_1]\!][\tau/\alpha] <: [\![T_2]\!]$,
- $\Gamma, \beta, h : [\![\Sigma_2]\!], x : [\![T_1]\!][\tau/\alpha] \vdash [\![C_{21}]\!] <: \alpha[\tau/\alpha]$, and
- $\Gamma, \beta, h : [\![\Sigma_2]\!], k : (x : [\![T_2]\!]) \rightarrow [\![C_{21}]\!] \vdash \alpha[\tau/\alpha] <: [\![C_2 2]\!]$.

for some τ. Since CPS-transformed types do not contain type variables, we have

- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!]$,
- $\Gamma, \beta, h : [\![\Sigma_2]\!] \vdash [\![T_1]\!] <: [\![T_2]\!]$,
- $\Gamma, \beta, h : [\![\Sigma_2]\!], x : [\![T_1]\!] \vdash [\![C_{21}]\!] <: \tau$, and
- $\Gamma, \beta, h : [\![\Sigma_2]\!], k : (x : [\![T_2]\!]) \rightarrow [\![C_{21}]\!] \vdash \tau <: [\![C_2 2]\!]$.

By 9, we have

- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!]$,
- $\Gamma, \beta \vdash [\![T_1]\!] <: [\![T_2]\!]$,
- $\Gamma, \beta, x : [\![T_1]\!] \vdash [\![C_{21}]\!] <: \tau$, and
- $\Gamma, \beta \vdash \tau <: [\![C_2 2]\!]$.

By Lemma 35 and 47, we have

- $\Gamma, \beta \vdash [\![\Sigma_2]\!] <: [\![\Sigma_1]\!]$,
- $\Gamma, \beta \vdash [\![T_1]\!] <: [\![T_2]\!]$, and
- $\Gamma, \beta, x : [\![T_1]\!] \vdash [\![C_{21}]\!] <: [\![C_22]\!]$.

By the IHs, we have

- $(\!|\Gamma|\!) \vdash \Sigma_2 <: \Sigma_1$,
- $(\!|\Gamma|\!) \vdash T_1 <: T_2$, and
- $(\!|\Gamma|\!), x : T_1 \vdash C_{21} <: C_{22}$.

Then we have the conclusion by (S-Embed) and (S-Comp).

Case $C_1 = \Sigma_1 \triangleright T_1 \ / \ (\forall x.C_{11}) \Rightarrow C_{12}$ and $C_2 = \Sigma_2 \triangleright T_2 \ / \ \square$: We have

- $[\![C_1]\!] = \forall \alpha.[\![\Sigma_1]\!] \to ((x : [\![T_1]\!]) \to [\![C_{11}]\!]) \to [\![C_{12}]\!]$ and
- $[\![C_2]\!] = \forall \beta.[\![\Sigma_2]\!] \to ([\![T_2]\!] \to \beta) \to \beta$.

By inversion, we have

- $\Gamma, \beta \vdash \tau$, and
- $\Gamma, \beta, h : [\![\Sigma_2]\!], k : (x : [\![T_2]\!]) \to \beta \vdash [\![C_{12}]\!][\tau/\alpha] <: \beta$.

for some τ. Since CPS-transformed types do not contain type variables, we have

- $\Gamma, \beta, h : [\![\Sigma_2]\!], k : (x : [\![T_2]\!]) \to \beta \vdash [\![C_{12}]\!] <: \beta$.

This is contradictory since $[\![C_{12}]\!]$ cannot be a type variable and thus there is no applicable rule.

3. By the IHs, and (S-Sig).

□

Theorem 67 (Backward type preservation (for open expressions))**.** *Assume that Γ is cps-wellformed.*

1. If $\Gamma \vdash [\![v]\!] : \tau$, then there exists T such that $(\!|\Gamma|\!) \vdash v : T$ and $\Gamma \vdash [\![T]\!] <: \tau$.

2. If $\Gamma \vdash [\![c]\!] : \tau$, then there exists C such that $(\!|\Gamma|\!) \vdash c : C$ and $\Gamma \vdash [\![C]\!] <: \tau$.

Proof. By simultaneous induction on the structure of v and c.

1. **Case $v = x$:** We have $[\![v]\!] = x$. By Lemma 48, we have *either*

1. $\vdash \Gamma$ and $\Gamma \vdash \{z : B \mid z = x\} <: \tau$ (if $\Gamma(x) = \{z : B \mid \phi\}$ for some z, B and ϕ)
2. $\vdash \Gamma$ and $\Gamma \vdash \Gamma(x) <: \tau$ (otherwise)

Case 1: By Lemma 65, we have $\vdash (\!|\Gamma|\!)$. Also, since $[\![\{z : B \mid \phi'\}]\!] = \{z : B \mid \phi'\}$ for any ϕ', we have

- $(\!|\Gamma|\!)(x) = \{z : B \mid \phi\}$ and
- $\Gamma \vdash [\![\{z : B \mid z = x\}]\!] <: \tau$.

Then, by (T-CVar), we have $(\!|\Gamma|\!) \vdash x : \{z : B \mid z = x\}$. Now we have the conclusion with $T = \{z : B \mid z = x\}$.

Case 2: By Lemma 65, we have $\vdash (\!|\Gamma|\!)$. Then, since $\Gamma(x) = \mathit{rmtv}(\Gamma)(x) = [\![(\!|\Gamma|\!)]\!](x) = [\![(\!|\Gamma|\!)(x)]\!]$ holds by Lemma 60, we have $\Gamma \vdash [\![(\!|\Gamma|\!)(x)]\!] <: \tau$. Also, by (T-Var), we have $(\!|\Gamma|\!) \vdash x : (\!|\Gamma|\!)(x)$. Now we have the conclusion with $T = (\!|\Gamma|\!)(x)$.

Case $v = p$: We have $[\![v]\!] = \mathit{cps}(p)$. By Lemma 48, we have

- $\vdash \Gamma$ and
- $\Gamma \vdash \mathit{ty}_{\mathit{cps}}([\![p]\!]) <: \tau$.

By Lemma 65, we have $\vdash (\!|\Gamma|\!)$. Then, by (T-Prim), we have $(\!|\Gamma|\!) \vdash p : \mathit{ty}(p)$. Also, by Assumption 52, we have $\Gamma \vdash [\![\mathit{ty}(p)]\!] <: \tau$. Now we have the conclusion with $T = \mathit{ty}(p)$.

Case $v = \mathbf{rec}(f^{(x:T_1) \to C_1}, x^{T_2}).c$: We have $[\![v]\!] = \mathbf{rec}(f : (x : [\![T_1]\!]) \to [\![C_1]\!], x : [\![T_2]\!]).[\![c]\!]$. By Lemma 48, we have

- $\Gamma, f : (x : [\![T_1]\!]) \to [\![C_1]\!], x : [\![T_1]\!] \vdash [\![c]\!] : [\![C_1]\!]$ and
- $\Gamma \vdash (x : [\![T_1]\!]) \to [\![C_1]\!] <: \tau$.

By the IH, we have

- $(\!|\Gamma|\!), f : (x : T_1) \to C_1, x : T_1 \vdash c : C_1'$ and
- $\Gamma, f : (x : [\![T_1]\!]) \to [\![C_1]\!], x : [\![T_1]\!] \vdash [\![C_1']\!] <: [\![C_1]\!]$

for some C_1'. By Lemma 66, we have

$$(\!|\Gamma|\!), f : (x : T_1) \to C_1, x : T_1 \vdash C_1' <: C_1 \ .$$

Then, by (T-CSub) and (T-Fun), we have

$$\Gamma \vdash \mathbf{rec}(f^{(x:T_1)\to C_1}, x^{T_2}).c : (x : T_1) \to C_1 \ .$$

Now we have the conclusion with $T = (x : T_1) \to C_1$.

2. **Case $c = \mathbf{return}\ v^T$:** We have $[\![c]\!] = \Lambda\alpha.\lambda h : \{\}.\lambda k : [\![T]\!] \to \alpha.k\ [\![v]\!]$. By Lemma 49, we have

- $\Gamma, \alpha, h : \{\}, k : [\![T]\!] \to \alpha \vdash k\ [\![v]\!] : \tau'$ and
- $\Gamma \vdash \forall\alpha.\{\} \to ([\![T]\!] \to \alpha) \to \tau' <: \tau$

for some τ'. By Lemma 48, we have

- $\Gamma, \alpha, h : \{\}, k : [\![T]\!] \to \alpha \vdash k : (y : \tau_1) \to \tau_2$,
- $\Gamma, \alpha, h : \{\}, k : [\![T]\!] \to \alpha \vdash [\![v]\!] : \tau_1$, and
- $\Gamma, \alpha, h : \{\}, k : [\![T]\!] \to \alpha \vdash \tau_2[[\![v]\!]/y] <: \tau'$

for some y, τ_1, and τ_2. By Lemma 39, we have

- $\Gamma, \alpha \vdash [\![v]\!] : \tau_1$.

Then, by the IH, we have

- $(\!|\Gamma|\!) \vdash v : T$ and
- $\Gamma \vdash [\![T]\!] <: \tau_1$.

Therefore, by (T-Ret), we have

- $(\!|\Gamma|\!) \vdash \mathbf{return}\ v : \{\} \triangleright T \ / \ \square$.

On the other hand, by Lemma 48, we have

- $\Gamma, \alpha, h : \{\}, k : [\![T]\!] \to \alpha \vdash [\![T]\!] \to \alpha <: (y : \tau_1) \to \tau_2$.

Then, By inversion, we have $\tau_2 = \alpha$. By inversion again, we have $\tau' = \alpha$. Therefore, we have

- $\Gamma \vdash \forall\alpha.\{\} \to ([\![T]\!] \to \alpha) \to \alpha <: \tau$,

that is,

- $\Gamma \vdash [\![\{\} \triangleright T \ / \ \square]\!] <: \tau$.

Now we have the conclusion with $C = \{\} \triangleright T \ / \ \square$.

Case $c = \mathbf{let}\ x = c_1^{\Sigma\triangleright T_1/\square}\ \mathbf{in}\ c_2^{\Sigma\triangleright T_2/\square}$: We have $[\![c]\!] = \Lambda\alpha.\lambda h : [\![\Sigma]\!].\lambda k : [\![T_2]\!] \to \alpha.[\![c_1]\!]\ \alpha\ h\ (\lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k)$. By Lemma 49, we have

(i) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha \vdash [\![c_1]\!]\ \alpha\ h\ (\lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k) : \tau'$ and
(ii) $\Gamma \vdash \forall\alpha.[\![\Sigma]\!] \to ([\![T_2]\!] \to \alpha) \to \tau' <: \tau$

for some τ'. By Lemma 50 with (i), we have

(iii) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha \vdash [\![c_1]\!] : \tau''$,
(iv) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha \vdash h : \tau_1$, and
(v) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha \vdash \lambda x : [\![T_1]\!].[\![c_2]\!]\ \alpha\ h\ k : \tau_2$

for some τ'', τ_1 and τ_2. By Lemma 48 with (iv) and (v) respectively, we have

(vi) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha \vdash [\![\Sigma]\!] <: \tau_1$,
(vii) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha, x : [\![T_1]\!] \vdash [\![c_2]\!]\ \alpha\ h\ k : \tau_3$, and
(viii) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha \vdash (x : [\![T_1]\!]) \to \tau_3 <: \tau_2$

for some τ_3. Then, by the second half of Lemma 50, we have

(ix) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha \vdash \tau'' <: \forall\beta.[\![\Sigma]\!] \to ((x : [\![T_1]\!]) \to \tau_3) \to \tau'$

where β is fresh.

On the other hand, by Lemma 50 with (vii), we have

(x) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha, x : [\![T_1]\!] \vdash [\![c_2]\!] : \tau_3'$,
(xi) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha, x : [\![T_1]\!] \vdash h : \tau_4$, and
(xii) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha, x : [\![T_1]\!] \vdash k : \tau_5$

for some τ_3', τ_4 and τ_5. By Lemma 48 with (xi) and (xii) respectively, we have

- $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha, x : [\![T_1]\!] \vdash [\![\Sigma]\!] <: \tau_4$ and
- $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha, x : [\![T_1]\!] \vdash [\![T_2]\!] \to \alpha <: \tau_5$.

Then, by the second half of Lemma 50, we have

(xiii) $\Gamma, \alpha, h : [\![\Sigma]\!], k : [\![T_2]\!] \to \alpha, x : [\![T_1]\!] \vdash \tau_3' <: \forall \gamma. [\![\Sigma]\!] \to ([\![T_2]\!] \to \alpha) \to \tau_3$

where γ is fresh.

By Lemma 39 with (iii), (ix), (x) and (xiii), we have

(xiv) $\Gamma, \alpha \vdash [\![c_1]\!] : \tau''$,

(xv) $\Gamma, \alpha \vdash \tau'' <: \forall \beta. [\![\Sigma]\!] \to ((x : [\![T_1]\!]) \to \tau_3) \to \tau'$

(xvi) $\Gamma, \alpha, x : [\![T_1]\!] \vdash [\![c_2]\!] : \tau_3'$

(xvii) $\Gamma, \alpha, x : [\![T_1]\!] \vdash \tau_3' <: \forall \gamma. [\![\Sigma]\!] \to ([\![T_2]\!] \to \alpha) \to \tau_3$.

Then, by the IHs of (xiv) and (xvi) respectively, we have

(xviii) $(\!|\Gamma|\!) \vdash c_1 : C_1$,

(xix) $\Gamma, \alpha \vdash [\![C_1]\!] <: \tau''$,

(xx) $(\!|\Gamma|\!), x : T_1 \vdash c_2 : C_2$, and

(xxi) $\Gamma, \alpha, x : [\![T_1]\!] \vdash [\![C_2]\!] <: \tau_3'$

for some C_1 and C_2. By Lemma 47 with "(xv) and (xix)" and "(xvii) and (xxi)" respectively, we have

(xxii) $\Gamma, \alpha \vdash [\![C_1]\!] <: \forall \beta. [\![\Sigma]\!] \to ((x : [\![T_1]\!]) \to \tau_3) \to \tau'$ and

(xxiii) $\Gamma, \alpha, x : [\![T_1]\!] \vdash [\![C_2]\!] <: \forall \gamma. [\![\Sigma]\!] \to ([\![T_2]\!] \to \alpha) \to \tau_3$.

By Lemma 63 with (xxiii), we have

- $C_1 = \Sigma_{11} \triangleright T_{11} \ / \ \square$ and
- $\tau_3 = \alpha$

for some Σ_{11} and T_{11}. Then, by Lemma 63 again with (xxii), we have

- $C_2 = \Sigma_{22} \triangleright T_{22} \ / \ \square$ and
- $\tau' = \alpha$

for some Σ_{22} and T_{22}. By inversion of (xxii), we have

- $\Gamma, \alpha, \beta \vdash [\![\Sigma]\!] <: [\![\Sigma_{11}]\!]$ and
- $\Gamma, \alpha, h : [\![\Sigma]\!], \beta \vdash [\![T_{11}]\!] <: [\![T_1]\!]$.

By Lemma 39, 40 and 66, we have

- $(\!|\Gamma|\!) \vdash \Sigma <: \Sigma_{11}$ and
- $(\!|\Gamma|\!) \vdash T_{11} <: T_1$.

Then, by subsumption on (xviii), we have

(xxiv) $(\!|\Gamma|\!) \vdash c_1 : \Sigma \triangleright T_1 \ / \ \square$.

In the same way, from (xxiii), we have

(xxv) $(\!|\Gamma|\!), x : T_1 \vdash c_2 : \Sigma \triangleright T_2 \ / \ \square$.

Therefore, by (T-LetP) with (xxiv) and (xxv), we have

$$(\!|\Gamma|\!) \vdash \mathbf{let}\ x = c_1\ \mathbf{in}\ c_2 : \Sigma \triangleright T_2 \ / \ \square \ .$$

Also, since $\tau' = \alpha$, (ii) implies

$$\Gamma \vdash [\![\Sigma \triangleright T_2 \ / \ \square]\!] <: \tau \ .$$

Now we have the conclusion with $C = \Sigma \triangleright T_2 \ / \ \square$.

Case $c = \mathbf{let}\ x = c_1^{\Sigma \triangleright T_1 / (\forall x. C_1) \Rightarrow C_2}\ \mathbf{in}\ c_2^{\Sigma \triangleright T_2 / (\forall z. C_0) \Rightarrow C_1}$**:** We have $[\![c]\!] = \Lambda \alpha. \lambda h : [\![\Sigma]\!]. \lambda k : (z : [\![T_2]\!]) \to [\![C_0]\!]. [\![c_1]\!]\ [\![C_2]\!]\ h\ (\lambda x : [\![T_1]\!]. [\![c_2]\!]\ [\![C_1]\!]\ h\ k)$. In the similar way to the previous case, we have

(i) $\Gamma \vdash \forall \alpha. [\![\Sigma]\!] \to ((z : [\![T_2]\!]) \to [\![C_0]\!]) \to \tau' <: \tau$,

(ii) $(\!|\Gamma|\!) \vdash c_1 : C_1$,

(iii) $\Gamma, \alpha \vdash [\![C_1]\!] <: \forall \beta. [\![\Sigma]\!] \to ((x : [\![T_1]\!]) \to \tau_3) \to \tau'$,

(iv) $(\!|\Gamma|\!), x : T_1 \vdash c_2 : C_2$, and

(v) $\Gamma, \alpha, x : [\![T_1]\!] \vdash [\![C_2]\!] <: \forall \gamma. [\![\Sigma]\!] \to ((z : [\![T_2]\!]) \to [\![C_0]\!]) \to \tau_3$

for some τ', τ_3, C_1 and C_2. By Lemma 64, we can assume that

- $C_1 = \Sigma_1 \triangleright T_{10} \;/\; (\forall x_1.C_{11}) \Rightarrow C_{12}$ and
- $C_2 = \Sigma_2 \triangleright T_{20} \;/\; (\forall x_2.C_{21}) \Rightarrow C_{22}$

for some $\Sigma_1, T_{10}, x_1, C_{11}, C_{12}, \Sigma_2, T_{20}, x_2, C_{21}$ and C_{22}. Then, by inversion of (iii), we have

(vi) $x_1 = x$

(vii) $\Gamma, \alpha, \beta \vdash [\![\Sigma]\!] <: [\![\Sigma_1]\!]$,

(viii) $\Gamma, \alpha, \beta, h : [\![\Sigma]\!] \vdash [\![T_{10}]\!] <: [\![T_1]\!]$,

(ix) $\Gamma, \alpha, \beta, h : [\![\Sigma]\!], x : [\![T_{10}]\!] \vdash \tau_3 <: [\![C_{11}]\!]$, and

(x) $\Gamma, \alpha, \beta, h : [\![\Sigma]\!], k : (x : [\![T_1]\!]) \to \tau_3 \vdash [\![C_{12}]\!] <: \tau'$.

By Lemma 39, 40 and 66 with (vii) and (viii) respectively, we have

(xi) $(\!|\Gamma|\!) \vdash \Sigma <: \Sigma_1$ and

(xii) $(\!|\Gamma|\!) \vdash T_{10} <: T_1$.

By subsumption on (ii) with (xi), we have

(xiii) $(\!|\Gamma|\!) \vdash c_1 : \Sigma \triangleright T_{10} \;/\; (\forall x_1.C_{11}) \Rightarrow C_{12}$.

On the other hand, by inversion of (v), we have

- $x_2 = z$
- $\Gamma, \alpha, x : [\![T_1]\!], \gamma \vdash [\![\Sigma]\!] <: [\![\Sigma_2]\!]$,
- $\Gamma, \alpha, x : [\![T_1]\!], \gamma, h : [\![\Sigma]\!] \vdash [\![T_{20}]\!] <: [\![T_2]\!]$,
- $\Gamma, \alpha, x : [\![T_1]\!], \gamma, h : [\![\Sigma]\!], z : [\![T_{20}]\!] \vdash [\![C_0]\!] <: [\![C_{21}]\!]$, and
- $\Gamma, \alpha, x : [\![T_1]\!], \gamma, h : [\![\Sigma]\!], k : (z : [\![T_2]\!]) \to [\![C_0]\!] \vdash [\![C_{22}]\!] <: \tau_3$.

By Lemma 36 with (viii), we have

(xiv) $\Gamma, \alpha, x : [\![T_{10}]\!], \gamma \vdash [\![\Sigma]\!] <: [\![\Sigma_2]\!]$,

(xv) $\Gamma, \alpha, x : [\![T_{10}]\!], \gamma, h : [\![\Sigma]\!] \vdash [\![T_{20}]\!] <: [\![T_2]\!]$,

(xvi) $\Gamma, \alpha, x : [\![T_{10}]\!], \gamma, h : [\![\Sigma]\!], z : [\![T_{20}]\!] \vdash [\![C_0]\!] <: [\![C_{21}]\!]$, and

(xvii) $\Gamma, \alpha, x : [\![T_{10}]\!], \gamma, h : [\![\Sigma]\!], k : (z : [\![T_2]\!]) \to [\![C_0]\!] \vdash [\![C_{22}]\!] <: \tau_3$.

By Lemma 39, 40 and 47 with (ix) and (xvii), we have

(xviii) $\Gamma, \alpha, x : [\![T_{10}]\!] \vdash [\![C_{22}]\!] <: [\![C_{11}]\!]$.

By Lemma 39, 40 and 66 with (xiv), (xv), (xvi) and (xviii), we have

- $(\!|\Gamma|\!), x : T_{10} \vdash \Sigma <: \Sigma_2$,
- $(\!|\Gamma|\!), x : T_{10} \vdash T_{20} <: T_2$,
- $(\!|\Gamma|\!), x : T_{10}, z : T_{20} \vdash C_0 <: C_{21}$, and
- $(\!|\Gamma|\!), x : T_{10} \vdash C_{22} <: C_{11}$.

Then, by Lemma 36 and subsumption on (iv), we have

(xix) $(\!|\Gamma|\!), x : T_{10} \vdash c_2 : \Sigma \triangleright T_2 \;/\; (\forall z.C_0) \Rightarrow C_{11}$.

Therefore, by (T-LetIp), we have

$$(\!|\Gamma|\!) \vdash \mathbf{let}\ x = c_1\ \mathbf{in}\ c_2 : \Sigma \triangleright T_2 \;/\; (\forall z.C_0) \Rightarrow C_{12} \ .$$

Also, by Lemma 39, 40, 35, and 47 with (i) and (x), we have

- $\Gamma \vdash \forall \alpha.[\![\Sigma]\!] \to ((z : [\![T_2]\!]) \to [\![C_0]\!]) \to [\![C_{12}]\!] <: \tau$,

that is,

$$\Gamma \vdash [\![\Sigma \triangleright T_2 \;/\; (\forall z.C_0) \Rightarrow C_{12}]\!] <: \tau \ .$$

Now we have the conclusion with $C = \Sigma \triangleright T_2 \;/\; (\forall z.C_0) \Rightarrow C_{12}$.

Case $c = v_1\ v_2$: We have $[\![c]\!] = [\![v_1]\!]\ [\![v_2]\!]$. By Lemma 48, we have

- $\Gamma \vdash [\![v_1]\!] : (x : \tau_1) \to \tau_2$,
- $\Gamma \vdash [\![v_2]\!] : \tau_1$, and
- $\Gamma \vdash \tau_2[[\![v_2]\!]/x] <: \tau$

for some x, τ_1 and τ_2. By the IHs, we have

- $(\!|\Gamma|\!) \vdash v_1 : T_1$,
- $(\!|\Gamma|\!) \vdash v_2 : T_2$,
- $\Gamma \vdash [\![T_1]\!] <: (x : \tau_1) \to \tau_2$, and

- $\Gamma \vdash [\![T_2]\!] <: \tau_1$

for some T_1 and T_2. By inversion, we have

- $T_1 = (x : T_{11}) \to C_{12}$,
- $\Gamma \vdash \tau_1 <: [\![T_{11}]\!]$, and
- $\Gamma, x : \tau_1 \vdash [\![C_{12}]\!] <: \tau_2$

for some T_{11} and C_{12}. By Lemma 47, we have $\Gamma \vdash [\![T_2]\!] <: [\![T_{11}]\!]$. Then, by Lemma 66, we have $(\!|\Gamma|\!) \vdash T_2 <: T_{11}$, and hence by (T-VSub) we have $(\!|\Gamma|\!) \vdash v_2 : T_{11}$. Therefore, by (T-App), we have $(\!|\Gamma|\!) \vdash v_1\ v2 : C_{12}[v_2/x]$.

On the other hand, by Lemma 41, we have $\Gamma \vdash [\![C_{12}]\!][[\![v_2]\!]/x] <: \tau_2[[\![v_2]\!]/x]$. Then, by Lemma 47, we have $\Gamma \vdash [\![C_{12}]\!][[\![v_2]\!]/x] <: \tau$.

Now we have the conclusion with $C = C_{12}[v_2/x]$.

Case $c = (\mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2)^{C'}$: We have $[\![c]\!] = (\mathbf{if}\ [\![v]\!]\ \mathbf{then}\ [\![c_1]\!]\ \mathbf{else}\ [\![c_2]\!] : [\![C']\!])$. By Lemma 48, we have

- $\Gamma \vdash \mathbf{if}\ [\![v]\!]\ \mathbf{then}\ [\![c_1]\!]\ \mathbf{else}\ [\![c_2]\!] : [\![C']\!]$ and
- $\Gamma \vdash [\![C']\!] <: \tau$.

By Lemma 48 again, we have

- $\Gamma \vdash [\![v]\!] : \{z : \mathrm{bool} \mid \phi\}$,
- $\Gamma, [\![v]\!] = \mathbf{true} \vdash [\![c_1]\!] : \tau'$,
- $\Gamma, [\![v]\!] = \mathbf{false} \vdash [\![c_2]\!] : \tau'$, and
- $\Gamma \vdash \tau' <: [\![C']\!]$

for some z, ϕ and τ'. By the IHs, we have

- $(\!|\Gamma|\!) \vdash v : \{z : \mathrm{bool} \mid \phi\}$,
- $(\!|\Gamma|\!), v = \mathbf{true} \vdash c_1 : C_1$,
- $(\!|\Gamma|\!), v = \mathbf{false} \vdash c_2 : C_2$,
- $\Gamma, [\![v]\!] = \mathbf{true} \vdash [\![C_1]\!] <: \tau'$, and
- $\Gamma, [\![v]\!] = \mathbf{false} \vdash [\![C_2]\!] <: \tau'$

for some C_1 and C_2. (Note that since v is of a refinement type, it holds that $[\![v]\!] = v$.) By Lemma 35 and 47, we have

- $\Gamma, [\![v]\!] = \mathbf{true} \vdash [\![C_1]\!] <: [\![C']\!]$ and
- $\Gamma, [\![v]\!] = \mathbf{false} \vdash [\![C_2]\!] <: [\![C']\!]$.

By Lemma 66, we have

- $(\!|\Gamma|\!), v = \mathbf{true} \vdash C_1 <: C'$ and
- $(\!|\Gamma|\!), v = \mathbf{false} \vdash C_2 <: C'$.

Then, by (T-CSub), we have

- $(\!|\Gamma|\!), v = \mathbf{true} \vdash c_1 : C'$ and
- $(\!|\Gamma|\!), v = \mathbf{false} \vdash c_2 : C'$.

Therefore by (T-If), we have $(\!|\Gamma|\!) \vdash \mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2 : C'$. Now we have the conclusion with $C = C'$.

Case $c = (\mathsf{op}^{\widetilde{A}}\ v)^{\Sigma \triangleright T/(\forall y.C_1) \Rightarrow C_2}$: We have $[\![c]\!] = \Lambda\alpha.\lambda h : [\![\Sigma]\!].\lambda k : (y : [\![T]\!]) \to [\![C_1]\!].h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y' : [\![T]\!].k\ y')$. By Lemma 49, we have

(i) $\Gamma, \alpha, h : [\![\Sigma]\!], k : (y : [\![T]\!]) \to [\![C_1]\!] \vdash h\#\mathsf{op}\ \widetilde{A}\ [\![v]\!]\ (\lambda y' : [\![T]\!].k\ y') : \tau'$ and
(ii) $\Gamma \vdash \forall\alpha.[\![\Sigma]\!] \to ((y : [\![T]\!]) \to [\![C_1]\!]) \to \tau' <: \tau$

for some τ'. (Below, we write $\Gamma_{\alpha,h,k}$ for $\Gamma, \alpha, h : [\![\Sigma]\!], k : (y : [\![T]\!]) \to [\![C_1]\!]$.)

By Lemma 48 with (i), we have

(iii) $\Gamma_{\alpha,h,k} \vdash \lambda y' : [\![T]\!].k\ y' : \tau_1$,
(iv) $\Gamma_{\alpha,h,k} \vdash [\![v]\!] : \tau_3$,
(v) $\widetilde{\Gamma_{\alpha,h,k} \vdash A : B}$,
(vi) $\Gamma_{\alpha,h,k} \vdash [\![\Sigma]\!] <: \{\ldots, \forall \widetilde{X : B}.\tau_5, \ldots\}$,
(vii) $\Gamma_{\alpha,h,k} \vdash \tau_5[\widetilde{A/X}] <: (x : \tau_3) \to \tau_4$,
(viii) $\Gamma_{\alpha,h,k} \vdash \tau_4[[\![v]\!]/x] <: \tau_1 \to \tau_2$, and

(ix) $\Gamma_{\alpha,h,k} \vdash \tau_2 <: \tau'$.

By Assumption 33 and 62 with (v), we have

- $(\!|\Gamma|\!) \vdash \widetilde{A : \widetilde{B}}$.

By inversion of (vi), we have

- $\Sigma = \{\ldots, \mathsf{op} : \forall \widetilde{X : \widetilde{B}}.(x_{\mathsf{op}} : T_{\mathsf{op}1}) \to ((y_{\mathsf{op}} : T_{\mathsf{op}2}) \to C_{\mathsf{op}1}) \to C_{\mathsf{op}2}, \ldots\}$ and
- $\Gamma_{\alpha,h,k}, \widetilde{X : \widetilde{B}} \vdash (x_{\mathsf{op}} : [\![T_{\mathsf{op}1}]\!]) \to ((y_{\mathsf{op}} : [\![T_{\mathsf{op}2}]\!]) \to [\![C_{\mathsf{op}1}]\!]) \to [\![C_{\mathsf{op}2}]\!] <: \tau_5$.

By repeatedly inverting this subtyping judgment with applying Lemma 42 with (v), Lemma 41 with (iv), and Lemma 47 with (vii), (viii) and (ix), we have

(x) $x = x_{\mathsf{op}}$,

(xi) $\Gamma_{\alpha,h,k} \vdash \tau_3 <: [\![T_{\mathsf{op}1}]\!][\widetilde{A/X}]$,

(xii) $\Gamma_{\alpha,h,k} \vdash \tau_1 <: (y_{\mathsf{op}} : [\![T_{\mathsf{op}2}]\!][\widetilde{A/X}][[\![v]\!]/x]) \to [\![C_{\mathsf{op}1}]\!][\widetilde{A/X}][[\![v]\!]/x]$, and

(xiii) $\Gamma_{\alpha,h,k} \vdash [\![C_{\mathsf{op}2}]\!][\widetilde{A/X}][[\![v]\!]/x] <: \tau'$.

By Lemma 39 with (iv), we have

- $\Gamma, \alpha \vdash [\![v]\!] : \tau_3$.

Then, by the IH, we have

(xiv) $(\!|\Gamma|\!) \vdash v : T_v$ and

(xv) $\Gamma, \alpha \vdash [\![T_v]\!] <: \tau_3$

for some T_v. By Lemma 47 with (xi) and (xv) (using Lemma 39), we have

- $\Gamma, \alpha \vdash [\![T_v]\!] <: [\![T_{\mathsf{op}1}]\!][\widetilde{A/X}]$.

Then, by Lemma 66, we have

- $(\!|\Gamma|\!) \vdash T_v <: T_{\mathsf{op}1}[\widetilde{A/X}]$

and hence, by (T-VSub) with (xiv), we have

- $(\!|\Gamma|\!) \vdash v : T_{\mathsf{op}1}[\widetilde{A/X}]$.

Also, by Lemma 44 and inversion, we have $\Gamma, \alpha \vdash [\![\Sigma]\!]$. Then by Lemma 65, we have $(\!|\Gamma|\!) \vdash \Sigma$. Therefore, by (T-Op), we have

$$(\!|\Gamma|\!) \vdash \mathsf{op}\ v : \Sigma \triangleright T_{\mathsf{op}2}[\widetilde{A/X}][v/x] \ / \ (\forall y_{\mathsf{op}}.C_{\mathsf{op}1}[\widetilde{A/X}][v/x]) \Rightarrow C_{\mathsf{op}2}[\widetilde{A/X}][v/x] \ .$$

On the other hand, by Lemma 48 with (iii), we have

(xvi) $\Gamma_{\alpha,h,k} \vdash (y' : [\![T]\!]) \to \tau_6 <: \tau_1$,

(xvii) $\Gamma_{\alpha,h,k}, y' : [\![T]\!] \vdash \tau_8[y'/y_0] <: \tau_6$,

(xviii) $\Gamma_{\alpha,h,k}, y' : [\![T]\!] \vdash (y : [\![T]\!]) \to [\![C_1]\!] <: (y_0 : \tau_7) \to \tau_8$, and

(xix) $\Gamma_{\alpha,h,k}, y' : [\![T]\!] \vdash y' : \tau_7$.

By inversion of (xviii), we have

- $y = y_0$ and
- $\Gamma_{\alpha,h,k}, y' : [\![T]\!], y : \tau_7 \vdash [\![C_1]\!] <: \tau_8$.

Then, by Lemma 41 with (xix), we have

- $\Gamma_{\alpha,h,k}, y' : [\![T]\!] \vdash [\![C_1]\!][y'/y] <: \tau_8[y'/y]$.

Then, by Lemma 47 with (xvii), we have

- $\Gamma_{\alpha,h,k}, y' : [\![T]\!] \vdash [\![C_1]\!][y'/y] <: \tau_6$

(Note that $y = y_0$). Then by (Sc-Fun), we have

- $\Gamma_{\alpha,h,k} \vdash (y' : [\![T]\!]) \to [\![C_1]\!][y'/y] <: (y' : [\![T]\!]) \to \tau_6$

and by α-renaming we have

- $\Gamma_{\alpha,h,k} \vdash (y : [\![T]\!]) \to [\![C_1]\!] <: (y' : [\![T]\!]) \to \tau_6$.

Then, by Lemma 47 with (xvi) and (xii), we have

(xx) $\Gamma_{\alpha,h,k} \vdash (y : [\![T]\!]) \to [\![C_1]\!] <: (y_{\mathsf{op}} : [\![T_{\mathsf{op}2}]\!][\widetilde{A/X}][[\![v]\!]/x]) \to [\![C_{\mathsf{op}1}]\!][\widetilde{A/X}][[\![v]\!]/x]$.

Therefore, by some subtyping rules with (xx) and (xiii), we have

- $\Gamma \vdash \forall \alpha. [\![\Sigma]\!] \rightarrow ((y_{\mathsf{op}} : [\![T_{\mathsf{op2}}]\!][\widetilde{A/X}][[\![v]\!]/x]) \rightarrow [\![C_{\mathsf{op1}}]\!][\widetilde{A/X}][[\![v]\!]/x]) \rightarrow [\![C_{\mathsf{op2}}]\!][\widetilde{A/X}][[\![v]\!]/x] <: \forall \alpha. [\![\Sigma]\!] \rightarrow ((y : [\![T]\!]) \rightarrow [\![C_1]\!]) \rightarrow \tau'$.

Then by Lemma 47 with (ii), we have

- $\Gamma \vdash \forall \alpha. [\![\Sigma]\!] \rightarrow ((y_{\mathsf{op}} : [\![T_{\mathsf{op2}}]\!][\widetilde{A/X}][[\![v]\!]/x]) \rightarrow [\![C_{\mathsf{op1}}]\!][\widetilde{A/X}][[\![v]\!]/x]) \rightarrow [\![C_{\mathsf{op2}}]\!][\widetilde{A/X}][[\![v]\!]/x] <: \tau$,

that is,

$$\Gamma \vdash [\![\Sigma \triangleright T_{\mathsf{op2}}[\widetilde{A/X}][v/x] \;/\; (\forall y_{\mathsf{op}}.C_{\mathsf{op1}}[\widetilde{A/X}][v/x]) \Rightarrow C_{\mathsf{op2}}[\widetilde{A/X}][v/x]]\!] <: \tau \ .$$

Now we have the conclusion with $C = \Sigma \triangleright T_{\mathsf{op2}}[\widetilde{A/X}][v/x] \;/\; (\forall y_{\mathsf{op}}.C_{\mathsf{op1}}[\widetilde{A/X}][v/x]) \Rightarrow C_{\mathsf{op2}}[\widetilde{A/X}][v/x]$.

Case $c = (\textbf{with } h \textbf{ handle } c)^C$: We have $[\![c]\!] = [\![c]\!]\ [\![C]\!]\ [\![h^{ops}]\!]\ [\![h^{ret}]\!]$ where

$$\begin{cases} h = \{\textbf{return } x_r^{T_r} \mapsto c_r, (\mathsf{op}_i^{\widetilde{X_i : B_i}}(x_i^{T_{i1}}, k_i^{(y_i : T_{i2}) \rightarrow C_{i1}}) \mapsto c_i)_i\} \\ [\![h^{ret}]\!] = \lambda x_r : [\![T_r]\!].[\![c_r]\!] \\ [\![h^{ops}]\!] = \{(\mathsf{op}_i = \Lambda \widetilde{X_i : B_i}.\lambda x_i : [\![T_{i1}]\!].\lambda k_i : (y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!].[\![c_i]\!])_i\} \end{cases}$$

By Lemma 50, we have

(i) $\Gamma \vdash [\![c]\!] : \tau'$,
(ii) $\Gamma \vdash [\![h^{ops}]\!] : \tau_1$, and
(iii) $\Gamma \vdash [\![h^{ret}]\!] : \tau_2$

for some τ', τ_1 and τ_2.

By Lemma 48 with (ii) and (iii), we have

(iv) $\left(\Gamma \vdash \Lambda \widetilde{X_i : B_i}.\lambda x_i : [\![T_{i1}]\!].\lambda k_i : (y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!].[\![c_i]\!] : \tau_i\right)_i$,
(v) $\Gamma \vdash \{(\mathsf{op}_i : \tau_i)_i\} <: \tau_1$,
(vi) $\Gamma, x_r : [\![T_r]\!] \vdash [\![c_r]\!] : \tau_3$, and
(vii) $\Gamma \vdash (x_r : [\![T_r]\!]) \rightarrow \tau_3 <: \tau_2$.

Then, by the second half of Lemma 50, we have

(viii) $\Gamma \vdash \tau' <: \forall \alpha. \{(\mathsf{op}_i : \tau_i)_i\} \rightarrow ((x_r : [\![T_r]\!]) \rightarrow \tau_3) \rightarrow \tau$

where α is fresh.

By the IH of (vi), we have

(ix) $(\!|\Gamma|\!), x_r : T_r \vdash c_r : C_r$ and
(x) $\Gamma, x_r : [\![T_r]\!] \vdash [\![C_r]\!] <: \tau_3$

for some C_r.

By repeatedly inverting (iv) and by Lemma 47, we have

(xi) $\left(\Gamma, \widetilde{X_i : B_i}, x_i : [\![T_{i1}]\!], k_i : (y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!] \vdash [\![c_i]\!] : \tau'_i\right)_i$ and
(xii) $\left(\Gamma \vdash \forall \widetilde{X_i : B_i}.(x_i : [\![T_{i1}]\!]) \rightarrow ((y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!]) \rightarrow \tau'_i <: \tau_i\right)_i$

for some τ'_i. By the IH of (xi), we have

(xiii) $\left((\!|\Gamma|\!), \widetilde{X_i : B_i}, x_i : T_{i1}, k_i : (y_i : T_{i2}) \rightarrow C_{i1} \vdash c_i : C_i\right)_i$ and
(xiv) $\left(\Gamma, \widetilde{X_i : B_i}, x_i : [\![T_{i1}]\!], k_i : (y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!] \vdash [\![C_i]\!] <: \tau'_i\right)_i$

for some C_i's. By (Sc-Fun) and (Sc-PPoly) with (xiv), we have

- $\left(\Gamma \vdash \forall \widetilde{X_i : B_i}.(x_i : [\![T_{i1}]\!]) \rightarrow ((y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!]) \rightarrow [\![C_i]\!] <: \forall \widetilde{X_i : B_i}.(x_i : [\![T_{i1}]\!]) \rightarrow ((y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!]) \rightarrow \tau'_i\right)_i$.

Then, by Lemma 47 with (xii), we have

(xv) $\left(\Gamma \vdash \forall \widetilde{X_i : B_i}.(x_i : [\![T_{i1}]\!]) \rightarrow ((y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!]) \rightarrow [\![C_i]\!] <: \tau_i\right)_i$.

Thus, by Lemma 47 and subtyping with (viii), (x) and (xv), we have

- $\Gamma \vdash \tau' <: \forall \alpha. \tau_s \rightarrow ((x_r : [\![T_r]\!]) \rightarrow [\![C_r]\!]) \rightarrow \tau$

where $\tau_s \stackrel{\text{def}}{=} \{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B_i}}.(x_i : [\![T_{i1}]\!]) \rightarrow ((y_i : [\![T_{i2}]\!]) \rightarrow [\![C_{i1}]\!]) \rightarrow [\![C_i]\!])_i\}$. Here, we define Σ to be $\{(\mathsf{op}_i : \forall \widetilde{X_i : \widetilde{B_i}}.(x_i : T_{i1}) \rightarrow ((y_i : T_{i2}) \rightarrow C_{i1}) \rightarrow C_i)_i\}$, Then, it holds that $\tau_s = [\![\Sigma]\!]$. That is, we have

(xvi) $\Gamma \vdash \tau' <: \forall \alpha.[\![\Sigma]\!] \rightarrow ((x_r : [\![T_r]\!]) \rightarrow [\![C_r]\!]) \rightarrow \tau$.

On the other hand, by the IH of (i), we have

(xvii) $(\!|\Gamma|\!) \vdash c : C_0$ and

(xviii) $\Gamma \vdash [\![C_0]\!] <: \tau'$

for some C_0. By Lemma 64, w.l.o.g., we can assume that $C_0 = \Sigma_0 \triangleright T_0 \ / \ (\forall x_0.C_{01}) \Rightarrow C_{02}$. Then, by Lemma 47 with (xvi) and (xviii), we have

- $\Gamma \vdash \forall \beta.[\![\Sigma_0]\!] \rightarrow ((x_0 : [\![T_0]\!]) \rightarrow [\![C_{01}]\!]) \rightarrow C_{02} <: \forall \alpha.[\![\Sigma]\!] \rightarrow ((x_r : [\![T_r]\!]) \rightarrow [\![C_r]\!]) \rightarrow \tau$.

Then, by inversion, we have

- $x_0 = x_r$,
- $\Gamma, \alpha \vdash [\![\Sigma]\!] <: [\![\Sigma_0]\!]$,
- $\Gamma, \alpha, h : [\![\Sigma]\!] \vdash [\![T_0]\!] <: [\![T_r]\!]$,
- $\Gamma, \alpha, h : [\![\Sigma]\!], x_r : [\![T_0]\!] \vdash [\![C_r]\!] <: [\![C_{01}]\!]$,

and

(xix) $\Gamma, \alpha, h : [\![\Sigma]\!], k : (x_r : [\![T_0]\!]) \rightarrow [\![C_{01}]\!] \vdash [\![C_{02}]\!] <: \tau$.

By Lemma 39, we have

- $\Gamma, \alpha \vdash [\![\Sigma]\!] <: [\![\Sigma_0]\!]$,
- $\Gamma, \alpha \vdash [\![T_0]\!] <: [\![T_r]\!]$, and
- $\Gamma, \alpha, x_r : [\![T_0]\!] \vdash [\![C_r]\!] <: [\![C_{01}]\!]$.

Then, by 66, we have

- $(\!|\Gamma|\!) \vdash \Sigma <: \Sigma_0$,
- $(\!|\Gamma|\!) \vdash T_0 <: T_r$, and
- $(\!|\Gamma|\!), x_r : T_0 \vdash C_r <: C_{01}$.

Therefore, by subsumption on (xvii), we have

(xx) $(\!|\Gamma|\!) \vdash c : \Sigma \triangleright T_r \ / \ (\forall x_r.C_r) \Rightarrow C_{02}$.

Thus, by (T-Hndl) with (ix), (xiii) and (xx), we have

$$(\!|\Gamma|\!) \vdash \textbf{with } h \textbf{ handle } c : C_{02} \ .$$

Also, by Lemma 39 and 40 with (xix), we have

$$\Gamma \vdash [\![C_{02}]\!] <: \tau \ .$$

Now we have the conclusion with $C = C_{02}$.

□

Corollary 68 (Backward type preservation (for closed expressions))**.**

- *If $\emptyset \vdash [\![v]\!] : \tau$, then there exists some T such that $\emptyset \vdash v : T$ and $\emptyset \vdash [\![T]\!] <: \tau$.*
- *If $\emptyset \vdash [\![c]\!] : \tau$, then there exists some C such that $\emptyset \vdash c : C$ and $\emptyset \vdash [\![C]\!] <: \tau$.*

Proof. Immediate from Theorem 67 since $\emptyset$ is obviously cps-wellformed. □