Modular Fully-Abstract Compilation by Approximate Back-Translation: Technical Appendix

Dominique Devriese[†] Marco Patrignani[*]

Frank Piessens[†]

[†] iMinds-Distrinet, KU Leuven, Belgium

first.last @ cs.kuleuven.be

[*] MPI-SWS, Saarbrücken, Germany

first.last@mpi-sws.org

Abstract

A compiler is *fully-abstract* if the compilation from source language programs to target language programs reflects and preserves behavioural equivalence. Such compilers have important security benefits, as they limit the power of an attacker interacting with the program in the target language to that of an attacker interacting with the program in the source language. Proving compiler full-abstraction is, however, rather complicated. A common proof technique is based on the *back-translation* of target-level program contexts to behaviourally-equivalent source-level contexts. However, constructing such a back-translation is problematic when the source language is not strong enough to embed an encoding of the target language. For instance, when compiling from the simply-typed λ-calculus (λ^{τ}) to the untyped λ-calculus (λ^{u}), the lack of recursive types in λ^{τ} prevents such a back-translation.

We propose a general and elegant solution for this problem. The key insight is that it suffices to construct an *approximate* back-translation. The approximation is only accurate up to a certain number of steps and conservative beyond that, in the sense that the context generated by the back-translation may diverge when the original would not, but not vice versa. Based on this insight, we describe a general technique for proving compiler full-abstraction and demonstrate it on a compiler from λ^{τ} to λ^{u}. The proof uses asymmetric cross-language logical relations and makes innovative use of step-indexing to express the relation between a context and its approximate back-translation. The proof extends easily to common compiler patterns such as modular compilation and it, to the best of our knowledge, it is the first compiler full abstraction proof to have been fully mechanised in Coq. We believe this proof technique can scale to challenging settings and enable simpler, more scalable proofs of compiler full-abstraction.

This report contains the technical appendix for a companion article by the same title.

Contents

1 The Source Language λ^{τ} **4**

1.1 Syntax . . . 4
1.2 Static Semantics . . . 4
1.3 Dynamic Semantics . . . 5
1.4 Program contexts . . . 6
1.5 Contextual equivalence . . . 7

2 The Target Language λ^{u} **8**

2.1 Syntax . . . 8
2.2 Well-scopedness . . . 8
2.3 Dynamic Semantics . . . 8
2.4 Program contexts . . . 10
2.5 Contextual equivalence . . . 11

3 Language and World Specifications **12**

3.1 General Language Specification . . . 12
3.2 General World Specification . . . 13
3.3 Language Specification for λ^{τ} . . . 14
3.4 Language Specification for λ^{u} . . . 15
3.5 World Specification . . . 16

4 Logical Relations **18**

5 Compiler **22**

5.1 Compiler definition: erase and protect . . . 22
5.2 Properties of erasure . . . 23
5.2.1 Compatibility lemmas . . . 24
5.3 Properties of dynamic type wrappers . . . 32
5.4 Contextual equivalence reflection . . . 37

6 Equivalence preservation and emulation **39**

6.1 n-approximate UVal . . . 39
6.2 EmulDV specification . . . 40
6.3 Upgrade/downgrade . . . 41
6.4 Injecting λ^{τ} into UVal . . . 49
6.5 Emulating λ^{u} in UVal . . . 63
6.6 Approximate back-translation . . . 76
6.7 Contextual equivalence preservation . . . 76

7 Compiler full abstraction **78**

8 Modular Full Abstraction **79**
8.1 Linking 79
8.2 Compiler 79
8.3 Additional Theorems and Proofs 79
8.3.1 Proofs about Modularity 82

Important note: as mentioned in the companion article, many of the logical relation definitions in this technical appendix are simplifications of the corresponding definitions in a paper by Hur and Dreyer [2011].

1 The Source Language λ^{τ}

This Section presents the syntax, static semantics and dynamic semantics of λ^{τ} (Sections 1.1 to 1.3, respectively). Then it defines program contexts and contextual equivalence for λ^{τ} (Sections 1.4 and 1.5). This calculus features $\mathtt{Unit}$ and $\mathtt{Bool}$ primitive types. We will use $\mathtt{b}$ to indicate values of those types and $\mathcal{B}$ to indicate those types when it is obvious.

1.1 Syntax

The syntax of λ^{τ} is presented below.

$$
\begin{array}{rl}
\mathit{Terms}^{\lambda^{\tau}} & \mathbf{t} ::= \mathtt{unit} \mid \mathtt{true} \mid \mathtt{false} \mid \lambda\mathbf{x} : \tau.\, \mathbf{t} \mid \mathbf{x} \mid \mathbf{t}\ \mathbf{t} \mid \mathbf{t}.\mathbf{1} \mid \mathbf{t}.\mathbf{2} \mid \langle \mathbf{t}, \mathbf{t} \rangle \\
& \quad \mid \mathrm{inl}\ \mathbf{t} \mid \mathrm{inr}\ \mathbf{t} \mid \mathrm{case}\ \mathbf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t} \mid \mathbf{t}; \mathbf{t} \\
& \quad \mid \mathrm{if}\ \mathbf{t}\ \mathrm{then}\ \mathbf{t}\ \mathrm{else}\ \mathbf{t} \mid \mathrm{fix}_{\tau \to \tau}\ \mathbf{t} \\
\mathit{Vals}^{\lambda^{\tau}} & \mathbf{v} ::= \mathtt{unit} \mid \mathtt{true} \mid \mathtt{false} \mid \lambda\mathbf{x} : \tau.\, \mathbf{t} \mid \langle \mathbf{v}, \mathbf{v} \rangle \mid \mathrm{inl}\ \mathbf{v} \mid \mathrm{inr}\ \mathbf{v} \\
\mathit{Types}^{\lambda^{\tau}} & \tau ::= \mathtt{Unit} \mid \mathtt{Bool} \mid \tau \to \tau \mid \tau \times \tau \mid \tau \uplus \tau \\
& \mathbf{\Gamma} ::= \emptyset \mid \mathbf{\Gamma}, \mathbf{x} : \tau \\
& \mathbb{E} ::= [\cdot] \mid \mathbb{E}\ \mathbf{t} \mid \mathbf{v}\ \mathbb{E} \mid \mathbb{E}.\mathbf{1} \mid \mathbb{E}.\mathbf{2} \mid \langle \mathbb{E}, \mathbf{t} \rangle \mid \langle \mathbf{v}, \mathbb{E} \rangle \\
& \quad \mid \mathrm{inl}\ \mathbb{E} \mid \mathrm{inr}\ \mathbb{E} \mid \mathrm{case}\ \mathbb{E}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2} \mid \mathbb{E}; \mathbf{t} \\
& \quad \mid \mathrm{if}\ \mathbb{E}\ \mathrm{then}\ \mathbf{t}\ \mathrm{else}\ \mathbf{t} \mid \mathrm{fix}_{\tau \to \tau}\ \mathbb{E}
\end{array}
$$

1.2 Static Semantics

The static semantics of λ^{τ} is given according to the following type judgements. There, $\mathbf{\Gamma}$ is the environment binding variables to types.

$$
\begin{array}{ll}
\mathbf{\Gamma} \vdash \diamond & \text{Well-formed environment } \mathbf{\Gamma} \\
\mathbf{\Gamma} \vdash \mathbf{t} : \tau & \text{Well-typed term } \mathbf{t} \text{ of type } \tau
\end{array}
$$

The type rules for λ^{τ} are given below.

$$
\frac{}{\emptyset \vdash \diamond}\ (\lambda^{\tau}\text{-Env-base})
\qquad
\frac{\mathbf{\Gamma} \vdash \diamond \qquad \mathbf{x} \notin \mathtt{dom}(\mathbf{\Gamma})}{\mathbf{\Gamma}, (\mathbf{x} : \tau) \vdash \diamond}\ (\lambda^{\tau}\text{-Env-ind})
\qquad
\frac{}{\mathbf{\Gamma} \vdash \mathtt{unit} : \mathtt{Unit}}\ (\lambda^{\tau}\text{-unit})
$$

$$
\frac{}{\mathbf{\Gamma} \vdash \mathtt{true} : \mathtt{Bool}}\ (\lambda^{\tau}\text{-true})
\qquad
\frac{}{\mathbf{\Gamma} \vdash \mathtt{false} : \mathtt{Bool}}\ (\lambda^{\tau}\text{-false})
\qquad
\frac{\mathbf{\Gamma} \vdash \diamond \qquad (\mathbf{x} : \tau) \in \mathbf{\Gamma}}{\mathbf{\Gamma} \vdash \mathbf{x} : \tau}\ (\lambda^{\tau}\text{-Type-var})
$$

$$
\frac{\mathbf{\Gamma}, (\mathbf{x} : \tau) \vdash \mathbf{t} : \tau'}{\mathbf{\Gamma} \vdash \lambda\mathbf{x} : \tau.\, \mathbf{t} : \tau \to \tau'}\ (\lambda^{\tau}\text{-Type-fun})
\qquad
\frac{\mathbf{\Gamma} \vdash \mathbf{t_1} : \tau_1 \qquad \mathbf{\Gamma} \vdash \mathbf{t_2} : \tau_2}{\mathbf{\Gamma} \vdash \langle \mathbf{t_1}, \mathbf{t_2} \rangle : \tau_1 \times \tau_2}\ (\lambda^{\tau}\text{-Type-pair})
$$

$$\frac{\Gamma \vdash \mathbf{t} : \tau' \to \tau \qquad \Gamma \vdash \mathbf{t'} : \tau'}{\Gamma \vdash \mathbf{t}\ \mathbf{t'} : \tau}\ (\lambda^\tau\text{-Type-app}) \qquad \frac{\Gamma \vdash \mathbf{t} : \tau_1 \times \tau_2}{\Gamma \vdash \mathbf{t}.\mathbf{1} : \tau_1}\ (\lambda^\tau\text{-Type-proj1}) \qquad \frac{\Gamma \vdash \mathbf{t} : \tau_1 \times \tau_2}{\Gamma \vdash \mathbf{t}.\mathbf{2} : \tau_2}\ (\lambda^\tau\text{-Type-proj2})$$

$$\frac{\Gamma \vdash \mathbf{t} : \tau}{\Gamma \vdash \mathrm{inl}\ \mathbf{t} : \tau \uplus \tau'}\ (\lambda^\tau\text{-Type-inl}) \qquad \frac{\Gamma \vdash \mathbf{t} : \tau}{\Gamma \vdash \mathrm{inr}\ \mathbf{t} : \tau' \uplus \tau}\ (\lambda^\tau\text{-Type-inr})$$

$$\frac{\Gamma \vdash \mathbf{t} : \tau_1 \uplus \tau_2 \quad \Gamma, (\mathbf{x_1} : \tau_1) \vdash \mathbf{t_1} : \tau \qquad \Gamma, (\mathbf{x_2} : \tau_2) \vdash \mathbf{t_2} : \tau}{\Gamma \vdash \mathrm{case}\ \mathbf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2} : \tau}\ (\lambda^\tau\text{-Type-case})$$

$$\frac{\Gamma \vdash \mathbf{t} : \mathtt{Bool} \quad \Gamma \vdash \mathbf{t_1} : \tau \qquad \Gamma \vdash \mathbf{t_2} : \tau}{\Gamma \vdash \mathrm{if}\ \mathbf{t}\ \mathrm{then}\ \mathbf{t_1}\ \mathrm{else}\ \mathbf{t_2} : \tau}\ (\lambda^\tau\text{-Type-if}) \qquad \frac{\Gamma \vdash \mathbf{t_1} : \mathtt{Unit} \qquad \Gamma \vdash \mathbf{t_2} : \tau}{\Gamma \vdash \mathbf{t_1}; \mathbf{t_2} : \tau}\ (\lambda^\tau\text{-Type-seq})$$

$$\frac{\Gamma \vdash \mathbf{t} : (\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)}{\Gamma \vdash \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{t} : \tau_1 \to \tau_2}\ (\lambda^\tau\text{-Type-fix})$$

1.3 Dynamic Semantics

The dynamic semantics of λ^τ is given as a relation $\hookrightarrow \subseteq \mathit{Terms}^{\lambda^\tau} \times \mathit{Terms}^{\lambda^\tau}$. The semantics relies on the definition of evaluation contexts $\mathbb{E}$, which model where the next primitive reduction is taking place. Additionally, the semantics relies on the (standard) capture-avoiding substitution function $\mathbf{t}[\mathbf{v}/\mathbf{x}]$ that replaces all occurrences of $\mathbf{x}$ in $\mathbf{t}$ with $\mathbf{v}$.

$$\begin{aligned}
\mathtt{true}[\mathbf{v}/\mathbf{x}] &= \mathtt{true} & \mathtt{false}[\mathbf{v}/\mathbf{x}] &= \mathtt{false}\\
\mathtt{unit}[\mathbf{v}/\mathbf{x}] &= \mathtt{unit} & \mathbf{x}[\mathbf{v}/\mathbf{x}] &= \mathbf{v}\\
\mathbf{y}[\mathbf{v}/\mathbf{x}] &= \mathbf{y} & &\text{if } \mathbf{x} \neq \mathbf{y}\\
(\lambda \mathbf{y} : \tau.\, \mathbf{t})[\mathbf{v}/\mathbf{x}] &= \lambda \mathbf{y} : \tau.\, \mathbf{t}[\mathbf{v}/\mathbf{x}] & &\text{if } \mathbf{x} \neq \mathbf{y} \text{ and } \mathbf{y} \notin \mathtt{FV}(\mathbf{v})\\
\langle \mathbf{t_1}, \mathbf{t_2} \rangle[\mathbf{v}/\mathbf{x}] &= \langle \mathbf{t_1}[\mathbf{v}/\mathbf{x}], \mathbf{t_2}[\mathbf{v}/\mathbf{x}] \rangle & \mathbf{t_1}\ \mathbf{t_2}[\mathbf{v}/\mathbf{x}] &= \mathbf{t_1}[\mathbf{v}/\mathbf{x}]\ \mathbf{t_2}[\mathbf{v}/\mathbf{x}]\\
\mathbf{t}.\mathbf{1}[\mathbf{v}/\mathbf{x}] &= \mathbf{t}[\mathbf{v}/\mathbf{x}].\mathbf{1} & \mathbf{t}.\mathbf{2}[\mathbf{v}/\mathbf{x}] &= \mathbf{t}[\mathbf{v}/\mathbf{x}].\mathbf{2}\\
(\mathrm{inl}\ \mathbf{t})[\mathbf{v}/\mathbf{x}] &= \mathrm{inl}\ (\mathbf{t}[\mathbf{v}/\mathbf{x}]) & (\mathrm{inr}\ \mathbf{t})[\mathbf{v}/\mathbf{x}] &= \mathrm{inr}\ (\mathbf{t}[\mathbf{v}/\mathbf{x}])\\
(\mathbf{t_1}; \mathbf{t_2})[\mathbf{v}/\mathbf{x}] &= \mathbf{t_1}[\mathbf{v}/\mathbf{x}]; \mathbf{t_2}[\mathbf{v}/\mathbf{x}]\\
(\mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{t})[\mathbf{v}/\mathbf{x}] &= \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{t}[\mathbf{v}/\mathbf{x}]
\end{aligned}$$

$$\begin{aligned}
(\mathrm{if}\ \mathbf{t}\ \mathrm{then}\ \mathbf{t_1}\ \mathrm{else}\ \mathbf{t_2})[\mathbf{v}/\mathbf{x}] &= \mathrm{if}\ \mathbf{t}[\mathbf{v}/\mathbf{x}]\ \mathrm{then}\ \mathbf{t_1}[\mathbf{v}/\mathbf{x}]\ \mathrm{else}\ \mathbf{t_2}[\mathbf{v}/\mathbf{x}]\\
\mathrm{case}\ \mathbf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2}[\mathbf{v}/\mathbf{x}] &= \quad \text{if } \mathbf{x_1} \neq \mathbf{x} \wedge \mathbf{x_2} \neq \mathbf{x} \wedge \mathbf{x_1}, \mathbf{x_2} \notin \mathtt{FV}(\mathbf{v})\\
&\quad \mathrm{case}\ \mathbf{t}[\mathbf{v}/\mathbf{x}]\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1}[\mathbf{v}/\mathbf{x}] \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2}[\mathbf{v}/\mathbf{x}]
\end{aligned}$$

Define a substitution mapping $\mathbf{m}$ as a mapping between a variable and a value, formally $\mathbf{m} ::= [\mathbf{v}/\mathbf{x}]$. A list of substitution mappings is denoted with γ. Define the application of a list of substitution mappings γ to a term $\mathbf{t}$ as follows:

$$\mathbf{t}(\emptyset) = \mathbf{t} \qquad\qquad \mathbf{t}([\mathbf{x}/\mathbf{v}]; \gamma) = \mathbf{t}[\mathbf{v}/\mathbf{x}](\gamma)$$

$$\frac{\mathbf{t} \hookrightarrow \mathbf{t}'}{\mathbb{E}[\mathbf{t}] \hookrightarrow \mathbb{E}[\mathbf{t}']}\ (\lambda^\tau\text{-Eval-ctx}) \qquad \frac{}{(\lambda \mathbf{x} : \tau.\, \mathbf{t})\ \mathbf{v} \hookrightarrow \mathbf{t}[\mathbf{v}/\mathbf{x}]}\ (\lambda^\tau\text{-Eval-beta}) \qquad \frac{}{\langle \mathbf{v_1}, \mathbf{v_2}\rangle.\mathbf{1} \hookrightarrow \mathbf{v_1}}\ (\lambda^\tau\text{-Eval-proj1})$$

$$\frac{}{\langle \mathbf{v_1}, \mathbf{v_2}\rangle.\mathbf{2} \hookrightarrow \mathbf{v_2}}\ (\lambda^\tau\text{-Eval-proj2})$$

$$\frac{}{\text{case inl } \mathbf{v} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2} \hookrightarrow \mathbf{t_1}[\mathbf{v}/\mathbf{x_1}]}\ (\lambda^\tau\text{-Eval-case-inl})$$

$$\frac{}{\text{case inr } \mathbf{v} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2} \hookrightarrow \mathbf{t_2}[\mathbf{v}/\mathbf{x_2}]}\ (\lambda^\tau\text{-Eval-case-inr})$$

$$\frac{\mathbf{v} \equiv \texttt{true} \Rightarrow \mathbf{t}' \equiv \mathbf{t_1} \qquad \mathbf{v} \equiv \texttt{false} \Rightarrow \mathbf{t}' \equiv \mathbf{t_2}}{\text{if } \mathbf{v} \text{ then } \mathbf{t_1} \text{ else } \mathbf{t_2} \hookrightarrow \mathbf{t}'}\ (\lambda^\tau\text{-Eval-if-v})$$

$$\frac{}{\texttt{unit}; \mathbf{t} \hookrightarrow \mathbf{t}}\ (\lambda^\tau\text{-Eval-seq-next})$$

$$\frac{}{\text{fix}_{\tau_1 \to \tau_2}\ (\lambda \mathbf{x} : \tau_1 \to \tau_2.\, \mathbf{t}) \hookrightarrow \mathbf{t}[(\lambda\ \mathbf{y} : \tau_1.\, \text{fix}_{\tau_1 \to \tau_2}\ (\lambda \mathbf{x} : \tau_1 \to \tau_2.\, \mathbf{t})\ \mathbf{y})/\mathbf{x}]}\ (\lambda^\tau\text{-Eval-fix})$$

1.4 Program contexts

We define program contexts $\mathfrak{C}$ as expressions with a single hole.

We define a typing judgement for program contexts $\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau$ by the following rules:

$$\frac{\vdash \mathfrak{C} : \mathbf{\Gamma}'', \tau'' \to (\mathbf{\Gamma}, \mathbf{x} : \tau'), \tau}{\vdash \lambda \mathbf{x} : \tau'.\, \mathfrak{C} : \mathbf{\Gamma}'', \tau'' \to \mathbf{\Gamma}, \tau' \to \tau}\ (\lambda^\tau\text{-Type-Ctx-Lam}) \qquad \frac{}{\vdash \cdot : \mathbf{\Gamma}, \tau \to \mathbf{\Gamma}, \tau}\ (\lambda^\tau\text{-Type-Ctx-Hole})$$

$$\frac{\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1 \qquad \mathbf{\Gamma} \vdash \mathbf{t_2} : \tau_2}{\vdash \langle \mathfrak{C}, \mathbf{t_2} \rangle : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1 \times \tau_2}\ (\lambda^\tau\text{-Type-Ctx-Pair1})$$

$$\frac{\mathbf{\Gamma} \vdash \mathbf{t_1} : \tau_1 \qquad \vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_2}{\vdash \langle \mathbf{t_1}, \mathfrak{C} \rangle : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1 \times \tau_2}\ (\lambda^\tau\text{-Type-Ctx-Pair2}) \qquad \frac{\vdash \mathfrak{C} : \mathbf{\Gamma}'', \tau'' \to \mathbf{\Gamma}, \tau}{\vdash \text{inl } \mathfrak{C} : \mathbf{\Gamma}'', \tau'' \to \mathbf{\Gamma}, \tau \uplus \tau'}\ (\lambda^\tau\text{-Type-Ctx-Inl})$$

$$\frac{\vdash \mathfrak{C} : \mathbf{\Gamma}'', \tau'' \to \mathbf{\Gamma}, \tau'}{\vdash \text{inr } \mathfrak{C} : \mathbf{\Gamma}'', \tau'' \to \mathbf{\Gamma}, \tau \uplus \tau'}\ (\lambda^\tau\text{-Type-Ctx-Inr}) \qquad \frac{\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1 \to \tau_2 \qquad \mathbf{\Gamma} \vdash \mathbf{t_2} : \tau_1}{\vdash \mathfrak{C}\ \mathbf{t_2} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_2}\ (\lambda^\tau\text{-Type-Ctx-App1})$$

$$\frac{\mathbf{\Gamma} \vdash \mathbf{t_1} : \tau_1 \to \tau_2 \qquad \vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1}{\vdash \mathbf{t_1}\ \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_2}\ (\lambda^\tau\text{-Type-Ctx-App2}) \qquad \frac{\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1 \uplus \tau_2}{\vdash \mathfrak{C}.\mathbf{1} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1}\ (\lambda^\tau\text{-Type-Ctx-Proj1})$$

$$\frac{\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1 \uplus \tau_2}{\vdash \mathfrak{C}.\mathbf{2} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_2}\ (\lambda^\tau\text{-Type-Ctx-Proj2})$$

$$\frac{\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_1 \uplus \tau_2 \qquad \mathbf{\Gamma}, \mathbf{x_1} : \tau_1 \vdash \mathbf{t_1} : \tau_3 \qquad \mathbf{\Gamma}, \mathbf{x_2} : \tau_2 \vdash \mathbf{t_2} : \tau_3}{\vdash \text{case } \mathfrak{C} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_3}\ (\lambda^\tau\text{-Type-Ctx-Case1})$$

$$\frac{\mathbf{\Gamma} \vdash \mathbf{t} : \tau_1 \uplus \tau_2 \qquad \vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to (\mathbf{\Gamma}, \mathbf{x_1} : \tau_1), \tau_3 \qquad \mathbf{\Gamma}, \mathbf{x_2} : \tau_2 \vdash \mathbf{t_2} : \tau_3}{\vdash \text{case } \mathbf{t} \text{ of inl } \mathbf{x_1} \mapsto \mathfrak{C} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_3} \ (\lambda^\tau\text{-Type-Ctx-Case2})$$

$$\frac{\mathbf{\Gamma} \vdash \mathbf{t} : \tau_1 \uplus \tau_2 \qquad \mathbf{\Gamma}, \mathbf{x_1} : \tau_1 \vdash \mathbf{t_1} : \tau_3 \qquad \vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to (\mathbf{\Gamma}, \mathbf{x_2} : \tau_2), \tau_3}{\vdash \text{case } \mathbf{t} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau_3} \ (\lambda^\tau\text{-Type-Ctx-Case3})$$

$$\frac{\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \texttt{Bool} \qquad \mathbf{\Gamma} \vdash \mathbf{t_1} : \tau \qquad \mathbf{\Gamma} \vdash \mathbf{t_2} : \tau}{\vdash \text{if } \mathfrak{C} \text{ then } \mathbf{t_1} \text{ else } \mathbf{t_2} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau} \ (\lambda^\tau\text{-Type-Ctx-If1})$$

$$\frac{\mathbf{\Gamma} \vdash \mathbf{t} : \texttt{Bool} \qquad \vdash \mathbb{E} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau \qquad \mathbf{\Gamma} \vdash \mathbf{t_2} : \tau}{\vdash \text{if } \mathbf{t} \text{ then } \mathfrak{C} \text{ else } \mathbf{t_2} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau} \ (\lambda^\tau\text{-Type-Ctx-If2})$$

$$\frac{\mathbf{\Gamma} \vdash \mathbf{t} : \texttt{Bool} \qquad \mathbf{\Gamma} \vdash \mathbf{t_1} : \tau \qquad \vdash \mathbb{E} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau}{\vdash \text{if } \mathbf{t} \text{ then } \mathbf{t_1} \text{ else } \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau} \ (\lambda^\tau\text{-Type-Ctx-If3})$$

$$\frac{\mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \texttt{Unit} \qquad \mathbf{\Gamma} \vdash \mathbf{t} : \tau}{\vdash \mathfrak{C}; \mathbf{t} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau} \ (\lambda^\tau\text{-Type-Ctx-Seq1}) \qquad \frac{\mathbf{\Gamma} \vdash \mathbf{t} : \texttt{Unit} \qquad \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau}{\vdash \mathbf{t}; \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau} \ (\lambda^\tau\text{-Type-Ctx-Seq2})$$

Lemma 1. *If* $\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau$ *and* $\mathbf{\Gamma}' \vdash \mathbf{t} : \tau'$, *then* $\mathbf{\Gamma} \vdash \mathfrak{C}[\mathbf{t}] : \tau$.

Proof. Easy induction on $\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau$. □

1.5 Contextual equivalence

Definition 1 (Termination)**.** *For a closed term* $\emptyset \vdash \mathbf{t} : \tau$, *we say that* $\mathbf{t}\Downarrow$ *iff there exists a* $\mathbf{v}$ *such that* $\mathbf{t} \hookrightarrow^* \mathbf{v}$.

Definition 2 (Contextual equivalence for λ^τ)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t_1} : \tau$ *and* $\mathbf{\Gamma} \vdash \mathbf{t_2} : \tau$, *then we define that* $\mathbf{\Gamma} \vdash \mathbf{t_1} \simeq_{ctx} \mathbf{t_2} : \tau$ *iff for all* $\mathfrak{C}$ *such that* $\vdash \mathfrak{C} : \mathbf{\Gamma}, \tau \to \emptyset, \tau'$, *we have that* $\mathfrak{C}[\mathbf{t_1}]\Downarrow$ *iff* $\mathfrak{C}[\mathbf{t_2}]\Downarrow$.

2 The Target Language λ^{u}

This Section presents the syntax and the dynamic semantics of λ^{u} (Section 2.1 and 2.3, respectively). It also define well-scopedness of terms (Section 2.2), program contexts (Section 2.4) and it defines contextual equivalence (Section 2.5).

2.1 Syntax

The syntax of λ^{u} is presented below.

$$
\begin{aligned}
\mathsf{t} ::= {} & \texttt{unit} \mid \texttt{true} \mid \texttt{false} \mid \lambda \mathsf{x}.\, \mathsf{t} \mid \mathsf{x} \mid \mathsf{t}\ \mathsf{t} \mid \mathsf{t.1} \mid \mathsf{t.2} \mid \langle \mathsf{t}, \mathsf{t} \rangle \mid \mathrm{inl}\ \mathsf{t} \mid \mathrm{inr}\ \mathsf{t} \mid \mathsf{wrong} \\
& \mid \mathrm{case}\ \mathsf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x}_1 \mapsto \mathsf{t} \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathsf{t} \mid \mathsf{t}; \mathsf{t} \mid \mathrm{if}\ \mathsf{t}\ \mathrm{then}\ \mathsf{t}\ \mathrm{else}\ \mathsf{t} \\
\mathsf{v} ::= {} & \texttt{unit} \mid \texttt{true} \mid \texttt{false} \mid \lambda \mathsf{x}.\, \mathsf{t} \mid \langle \mathsf{v}, \mathsf{v} \rangle \mid \mathrm{inl}\ \mathsf{v} \mid \mathrm{inr}\ \mathsf{v} \\
\Gamma ::= {} & \emptyset \mid \Gamma, \mathsf{x} \\
\mathbb{E} ::= {} & [\cdot] \mid \mathbb{E}\ \mathsf{t} \mid \mathsf{v}\ \mathbb{E} \mid \mathbb{E}.1 \mid \mathbb{E}.2 \mid \langle \mathbb{E}, \mathsf{t} \rangle \mid \langle \mathsf{v}, \mathbb{E} \rangle \\
& \mid \mathrm{inl}\ \mathbb{E} \mid \mathrm{inr}\ \mathbb{E} \mid \mathrm{case}\ \mathbb{E}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathsf{t}_2 \mid \mathbb{E}; \mathsf{t} \mid \mathrm{if}\ \mathbb{E}\ \mathrm{then}\ \mathsf{t}\ \mathrm{else}\ \mathsf{t}
\end{aligned}
$$

2.2 Well-scopedness

We define a well-scopedness judgement for λ^{u} in terms of contexts Γ that are a list of in-scope variables.

The rules for the well-scopedness judgement are unsurprising:

$$
\frac{}{\Gamma \vdash \mathsf{b}}\ (\lambda^{u}\text{-Wf-Base})
\qquad
\frac{\Gamma, \mathsf{x} \vdash \mathsf{t}}{\Gamma \vdash \lambda \mathsf{x}.\, \mathsf{t}}\ (\lambda^{u}\text{-Wf-Lam})
\qquad
\frac{\mathsf{x} \in \Gamma}{\Gamma \vdash \mathsf{x}}\ (\lambda^{u}\text{-Wf-Var})
\qquad
\frac{\Gamma \vdash \mathsf{t}_1 \quad \Gamma \vdash \mathsf{t}_2}{\Gamma \vdash \langle \mathsf{t}_1, \mathsf{t}_2 \rangle}\ (\lambda^{u}\text{-Wf-Pair})
$$

$$
\frac{\Gamma \vdash \mathsf{t}}{\Gamma \vdash \mathrm{inl}\ \mathsf{t}}\ (\lambda^{u}\text{-Wf-Inl})
\qquad
\frac{\Gamma \vdash \mathsf{t}}{\Gamma \vdash \mathrm{inr}\ \mathsf{t}}\ (\lambda^{u}\text{-Wf-Inr})
\qquad
\frac{\Gamma \vdash \mathsf{t}_1 \quad \Gamma \vdash \mathsf{t}_2}{\Gamma \vdash \mathsf{t}_1\ \mathsf{t}_2}\ (\lambda^{u}\text{-Wf-App})
\qquad
\frac{\Gamma \vdash \mathsf{t}}{\Gamma \vdash \mathsf{t.1}}\ (\lambda^{u}\text{-Wf-Proj1})
$$

$$
\frac{\Gamma \vdash \mathsf{t}}{\Gamma \vdash \mathsf{t.2}}\ (\lambda^{u}\text{-Wf-Proj2})
\qquad
\frac{\Gamma \vdash \mathsf{t} \quad \Gamma, \mathsf{x}_1 \vdash \mathsf{t}_1 \quad \Gamma, \mathsf{x}_2 \vdash \mathsf{t}_2}{\Gamma \vdash \mathrm{case}\ \mathsf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathsf{t}_2}\ (\lambda^{u}\text{-Wf-Case})
\qquad
\frac{}{\Gamma \vdash \mathsf{wrong}}\ (\lambda^{u}\text{-Wf-Wrong})
$$

$$
\frac{\Gamma \vdash \mathsf{t} \quad \Gamma \vdash \mathsf{t}_1 \quad \Gamma \vdash \mathsf{t}_2}{\Gamma \vdash \mathrm{if}\ \mathsf{t}\ \mathrm{then}\ \mathsf{t}_1\ \mathrm{else}\ \mathsf{t}_2}\ (\lambda^{u}\text{-Wf-If})
\qquad
\frac{\Gamma \vdash \mathsf{t}_1 \quad \Gamma \vdash \mathsf{t}_2}{\Gamma \vdash \mathsf{t}_1; \mathsf{t}_2}\ (\lambda^{u}\text{-Wf-Seq})
$$

2.3 Dynamic Semantics

The dynamic semantics of λ^{u} is given as a relation $\hookrightarrow \subseteq \mathit{Terms}^{\lambda^{u}} \times \mathit{Terms}^{\lambda^{u}}$. The semantics relies on the definition of evaluation contexts $\mathbb{E}$, which model where the next primitive reduction is taking place. Additionally, the semantics relies on the capture-avoiding substitution function $\mathsf{t}[\mathsf{v}/\mathsf{x}]$ that replaces all occurrences of x in t with v.

$$
\begin{array}{rlrl}
\mathtt{true}[\mathsf{v}/\mathsf{x}] = \mathtt{true} & & \mathtt{false}[\mathsf{v}/\mathsf{x}] = \mathtt{false} & \\
\mathtt{unit}[\mathsf{v}/\mathsf{x}] = \mathtt{unit} & & \mathsf{x}[\mathsf{v}/\mathsf{x}] = \mathsf{v} & \\
\mathsf{y}[\mathsf{v}/\mathsf{x}] = \mathsf{y} & & \text{if } \mathsf{x} \neq \mathsf{y} & \\
(\lambda \mathsf{y}.\, \mathsf{t})[\mathsf{v}/\mathsf{x}] = \lambda \mathsf{y}.\, \mathsf{t}[\mathsf{v}/\mathsf{x}] & & \text{if } \mathsf{x} \neq \mathsf{y} \text{ and } \mathsf{y} \notin \mathtt{FV}(\mathsf{v}) & \\
\langle \mathsf{t}_1, \mathsf{t}_2 \rangle[\mathsf{v}/\mathsf{x}] = \langle \mathsf{t}_1[\mathsf{v}/\mathsf{x}], \mathsf{t}_2[\mathsf{v}/\mathsf{x}] \rangle & & \mathsf{t}_1\ \mathsf{t}_2[\mathsf{v}/\mathsf{x}] = \mathsf{t}_1[\mathsf{v}/\mathsf{x}]\ \mathsf{t}_2[\mathsf{v}/\mathsf{x}] & \\
\mathsf{t.1}[\mathsf{v}/\mathsf{x}] = \mathsf{t}[\mathsf{v}/\mathsf{x}].1 & & \mathsf{t.2}[\mathsf{v}/\mathsf{x}] = \mathsf{t}[\mathsf{v}/\mathsf{x}].2 & \\
\mathsf{wrong}[\mathsf{v}/\mathsf{x}] = \mathsf{wrong} & & & \\
\text{inl } \mathsf{t}[\mathsf{v}/\mathsf{x}] = \text{inl } (\mathsf{t}[\mathsf{v}/\mathsf{x}]) & & \text{inr } \mathsf{t}[\mathsf{v}/\mathsf{x}] = \text{inr } (\mathsf{t}[\mathsf{v}/\mathsf{x}]) & \\
(\mathsf{t}_1; \mathsf{t}_2)[\mathsf{v}/\mathsf{x}] = \mathsf{t}_1[\mathsf{v}/\mathsf{x}]; \mathsf{t}_2[\mathsf{v}/\mathsf{x}] & & (\text{if } \mathtt{t} \text{ then } \mathsf{t}_1 \text{ else } \mathsf{t}_2)[\mathsf{v}/\mathsf{x}] = \text{if } \mathsf{t}[\mathsf{v}/\mathsf{x}] \text{ then } \mathsf{t}_1[\mathsf{v}/\mathsf{x}] \text{ else } \mathsf{t}_2[\mathsf{v}/\mathsf{x}] &
\end{array}
$$

$$
\begin{array}{l}
\text{case } \mathsf{t} \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2[\mathsf{v}/\mathsf{x}] = \quad \text{if } \mathsf{x}_1 \neq \mathsf{x} \wedge \mathsf{x}_2 \neq \mathsf{x} \wedge \mathsf{x}_1, \mathsf{x}_2 \notin \mathtt{FV}(\mathsf{v}) \\
\qquad \text{case } \mathsf{t}[\mathsf{v}/\mathsf{x}] \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1[\mathsf{v}/\mathsf{x}] \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2[\mathsf{v}/\mathsf{x}]
\end{array}
$$

Define a substitution mapping m as a mapping between a variable and a value, formally $\mathsf{m} ::= [\mathsf{x}/\mathsf{v}]$. A list of substitution mappings is denoted with γ. Define the application of a list of substitution mappings γ to a term t as follows:

$$
\mathsf{t}(\emptyset) = \mathsf{t} \qquad\qquad \mathsf{t}([\mathsf{x}/\mathsf{v}]; \gamma) = \mathsf{t}[\mathsf{v}/\mathsf{x}](\gamma)
$$

$$\frac{\mathsf{t} \hookrightarrow \mathsf{t}'}{\mathbb{E}[\mathsf{t}] \hookrightarrow \mathbb{E}[\mathsf{t}']}\ (\lambda^{\mathsf{u}}\text{-Eval-ctx}) \qquad \frac{\mathbb{E} \neq [\cdot]}{\mathbb{E}[\mathsf{wrong}] \hookrightarrow \mathsf{wrong}}\ (\lambda^{\mathsf{u}}\text{-Eval-ctx-wrong}) \qquad \frac{}{(\lambda \mathsf{x}.\, \mathsf{t})\ \mathsf{v} \hookrightarrow \mathsf{t}[\mathsf{v}/\mathsf{x}]}\ (\lambda^{\mathsf{u}}\text{-Eval-beta})$$

$$\frac{}{\langle \mathsf{v}_1, \mathsf{v}_2 \rangle.1 \hookrightarrow \mathsf{v}_1}\ (\lambda^{\mathsf{u}}\text{-Eval-proj1}) \qquad \frac{}{\langle \mathsf{v}_1, \mathsf{v}_2 \rangle.2 \hookrightarrow \mathsf{v}_2}\ (\lambda^{\mathsf{u}}\text{-Eval-proj2})$$

$$\frac{}{\mathrm{case\ inl}\ \mathsf{v}\ \mathrm{of\ inl}\ \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathsf{t}_2 \hookrightarrow \mathsf{t}_1[\mathsf{v}/\mathsf{x}_1]}\ (\lambda^{\mathsf{u}}\text{-Eval-case-inl})$$

$$\frac{}{\mathrm{case\ inr}\ \mathsf{v}\ \mathrm{of\ inl}\ \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathsf{t}_2 \hookrightarrow \mathsf{t}_2[\mathsf{v}/\mathsf{x}_2]}\ (\lambda^{\mathsf{u}}\text{-Eval-case-inr}) \qquad \frac{\nexists \mathsf{t}.\ \mathsf{v} \equiv \lambda \mathsf{x}.\, \mathsf{t}}{\mathsf{v}\ \mathsf{v}' \hookrightarrow \mathsf{wrong}}\ (\lambda^{\mathsf{u}}\text{-Eval-beta-w})$$

$$\frac{\nexists \mathsf{v}'.\ \mathsf{v} \equiv \mathrm{inl}\ \mathsf{v}' \lor \mathsf{v} \equiv \mathrm{inr}\ \mathsf{v}'}{\mathrm{case}\ \mathsf{v}\ \mathrm{of\ inl}\ \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathsf{t}_2 \hookrightarrow \mathsf{wrong}}\ (\lambda^{\mathsf{u}}\text{-Eval-case-w})$$

$$\frac{\mathsf{j} \in \{1, 2\} \qquad \nexists \mathsf{v}_1, \mathsf{v}_2.\ \mathsf{v} \equiv \langle \mathsf{v}_1, \mathsf{v}_2 \rangle}{\mathsf{v}.\mathsf{j} \hookrightarrow \mathsf{wrong}}\ (\lambda^{\mathsf{u}}\text{-Eval-proj-wrong})$$

$$\frac{\begin{array}{c}\mathsf{v} \equiv \mathtt{true} \Rightarrow \mathsf{t}' \equiv \mathsf{t}_1 \qquad \mathsf{v} \equiv \mathtt{false} \Rightarrow \mathsf{t}' \equiv \mathsf{t}_2 \\ (\mathsf{v} \not\equiv \mathtt{true} \land \mathsf{v} \not\equiv \mathtt{false}) \Rightarrow \mathsf{t}' \equiv \mathsf{wrong}\end{array}}{\mathrm{if}\ \mathsf{v}\ \mathrm{then}\ \mathsf{t}_1\ \mathrm{else}\ \mathsf{t}_2 \hookrightarrow \mathsf{t}'}\ (\lambda^{\mathsf{u}}\text{-Eval-if-v}) \qquad \frac{}{\mathtt{unit}; \mathsf{t} \hookrightarrow \mathsf{t}}\ (\lambda^{\mathsf{u}}\text{-Eval-seq-next})$$

$$\frac{\mathsf{v} \not\equiv \mathtt{unit}}{\mathsf{v}; \mathsf{t} \hookrightarrow \mathsf{wrong}}\ (\lambda^{\mathsf{u}}\text{-Eval-seq-wrong})$$

Since λ^{u} is untyped, some reduction can result in a stuck term wrong, e.g., applying a non-lambda value to an argument (Rule λ^{u}-Eval-beta-w) or projecting over a function (Rule λ^{u}-Eval-proj-wrong).

2.4 Program contexts

We define program contexts $\mathfrak{C}$ as expressions with a single hole.

We define a well-scopedness judgement for program contexts $\mathfrak{C} : \Gamma' \to \Gamma$ inductively by the following rules:

$$\frac{\vdash \mathfrak{C} : \Gamma' \to (\Gamma, x)}{\vdash \lambda x.\, \mathfrak{C} : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Lam}) \qquad \frac{}{\vdash \cdot : \Gamma \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Hole}) \qquad \frac{\vdash \mathfrak{C} : \Gamma' \to \Gamma \quad \Gamma \vdash t_2}{\vdash \langle \mathfrak{C}, t_2 \rangle : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Pair1})$$

$$\frac{\Gamma \vdash t_1 \quad \vdash \mathfrak{C} : \Gamma' \to \Gamma}{\vdash \langle t_1, \mathfrak{C} \rangle : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Pair2}) \qquad \frac{\vdash \mathfrak{C} : \Gamma' \to \Gamma}{\vdash \mathrm{inl}\ \mathfrak{C} : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Inl}) \qquad \frac{\vdash \mathfrak{C} : \Gamma' \to \Gamma}{\vdash \mathrm{inr}\ \mathfrak{C} : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Inr})$$

$$\frac{\vdash \mathfrak{C} : \Gamma' \to \Gamma \quad \Gamma \vdash t_2}{\vdash \mathfrak{C}\ t_2 : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-App1}) \qquad \frac{\Gamma \vdash t_1 \quad \vdash \mathfrak{C} : \Gamma' \to \Gamma}{\vdash t_1\ \mathfrak{C} : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-App2}) \qquad \frac{\vdash \mathfrak{C} : \Gamma' \to \Gamma}{\vdash \mathfrak{C}.1 : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Proj1})$$

$$\frac{\vdash \mathfrak{C} : \Gamma' \to \Gamma}{\vdash \mathfrak{C}.2 : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Proj2}) \qquad \frac{\vdash \mathfrak{C} : \Gamma' \to \Gamma \quad \Gamma, x_1 \vdash t_1 \quad \Gamma, x_2 \vdash t_2}{\vdash \mathrm{case}\ \mathfrak{C}\ \mathrm{of}\ \mathrm{inl}\ x_1 \mapsto t_1 \mid \mathrm{inr}\ x_2 \mapsto t_2 : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Case1})$$

$$\frac{\Gamma \vdash t \quad \vdash \mathfrak{C} : \Gamma' \to (\Gamma, x_1) \quad \Gamma, x_2 \vdash t_2}{\vdash \mathrm{case}\ t\ \mathrm{of}\ \mathrm{inl}\ x_1 \mapsto \mathfrak{C} \mid \mathrm{inr}\ x_2 \mapsto t_2 : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Case2})$$

$$\frac{\Gamma \vdash t \quad \Gamma, x_1 \vdash t_1 \quad \vdash \mathfrak{C} : \Gamma' \to (\Gamma, x_2)}{\vdash \mathrm{case}\ t\ \mathrm{of}\ \mathrm{inl}\ x_1 \mapsto t_1 \mid \mathrm{inr}\ x_2 \mapsto \mathfrak{C} : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Wf-Ctx-Case3})$$

$$\frac{\vdash \mathfrak{C} : \Gamma' \to \Gamma \quad \Gamma \vdash t_1 \quad \Gamma \vdash t_2}{\vdash \mathrm{if}\ \mathfrak{C}\ \mathrm{then}\ t_1\ \mathrm{else}\ t_2 : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Type-Ctx-If1}) \qquad \frac{\Gamma \vdash t \quad \vdash \mathbb{E} : \Gamma' \to \Gamma \quad \Gamma \vdash t_2}{\vdash \mathrm{if}\ t\ \mathrm{then}\ \mathfrak{C}\ \mathrm{else}\ t_2 : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Type-Ctx-If2})$$

$$\frac{\Gamma \vdash t \quad \Gamma \vdash t_1 \quad \vdash \mathbb{E} : \Gamma' \to \Gamma}{\vdash \mathrm{if}\ t\ \mathrm{then}\ t_1\ \mathrm{else}\ \mathfrak{C} : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Type-Ctx-If3}) \qquad \frac{\mathfrak{C} : \Gamma' \to \Gamma \quad \Gamma \vdash t}{\vdash \mathfrak{C}; t : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Type-Ctx-Seq1})$$

$$\frac{\Gamma \vdash t \quad \mathfrak{C} : \Gamma' \to \Gamma}{\vdash t; \mathfrak{C} : \Gamma' \to \Gamma}\ (\lambda^{u}\text{-Type-Ctx-Seq2})$$

2.5 Contextual equivalence

Definition 3 (Contextual equivalence for λ^{u}). *If $\Gamma \vdash t_1$ and $\Gamma \vdash t_2$, then we define that $\Gamma \vdash t_1 \simeq_{ctx} t_2$ iff for all $\mathfrak{C}$ such that $\vdash \mathfrak{C} : \Gamma \to \emptyset$, we have that $\mathfrak{C}[t_1]\Downarrow$ iff $\mathfrak{C}[t_2]\Downarrow$.*

3 Language and World Specifications

This Section defines general language and world specifications *LSpec* and *WSpec* (Section 3.1 and Section 3.2, respectively). Then, a concrete language specifications for both λ^{τ} and λ^{u} is provided (Sections 3.3 to 3.4), as well as a concrete world specification (Section 3.5).

3.1 General Language Specification

The general language specification is presented below.

$$
\begin{aligned}
\mathit{LSpec} \stackrel{\mathsf{def}}{=} \{&\mathsf{Val}, \mathsf{Ter}, \mathsf{Con}, \mathsf{Conf}, \\
&\mathsf{plugv}, \mathsf{plugc}, \mathsf{step}, \mathsf{oftype}, \mathsf{bool}, \\
&\mathsf{unit}, \mathsf{pair}, \mathsf{appl}, \mathsf{inl}, \mathsf{inr} \mid \\
&\mathsf{Val}, \mathsf{Ter}, \mathsf{Con}, \mathsf{Conf} \in \mathit{Set} && \wedge\, \mathsf{plugv} \in \mathsf{Val} \times \mathsf{Con} \to \mathcal{P}(\mathsf{Conf}) \\
&\wedge\, \mathsf{plugc} \in \mathsf{Ter} \times \mathsf{Con} \to \mathcal{P}(\mathsf{Conf}) && \wedge\, \mathsf{step} \in \mathsf{Conf} \to \mathsf{Conf} \uplus \{\mathtt{halt}, \mathtt{fail}\} \\
&\wedge\, \mathsf{oftype} \in \mathit{Types}^{\lambda^{\tau}} \to \mathcal{P}(\mathsf{Val}) && \wedge\, \mathsf{unit} \in \mathtt{Unit} \to \mathcal{P}(\mathsf{Val}) \\
&\wedge\, \mathsf{bool} \in \mathtt{Bool} \to \mathcal{P}(\mathsf{Val}) && \wedge\, \mathsf{pair} \in \mathsf{Val} \times \mathsf{Val} \to \mathcal{P}(\mathsf{Val}) \\
&\wedge\, \mathsf{appl} \in \mathsf{Val} \times \mathsf{Val} \to \mathcal{P}(\mathsf{Ter}) \\
&\wedge\, \mathsf{inl} \in \mathsf{Val} \to \mathcal{P}(\mathsf{Val}) && \wedge\, \mathsf{inr} \in \mathsf{Val} \to \mathcal{P}(\mathsf{Val})\}
\end{aligned}
$$

For a language to implement the language specifications, it must provide values (Val), terms (Ter), continuations (also known as contexts, Con) and configurations (Conf). Then, it must provide functions to plug a value in a continuation (plugv), to plug a term in a continuation (plugc), to perform a reduction step (step), to identify the values of a type (oftype), to identify primitive values (base), to build pairs (pair) and to apply functions to arguments (appl). This specification will need to be enriched in case either the source or the target languages are enriched (i.e., when references are added, memories must be modelled).

Define a configuration $t \in \mathsf{Conf}$ performing k reduction, denoted as $t \overset{\mathrm{k}}{\hookrightarrow} t'$ steps as follows:

$$
\begin{aligned}
&t \overset{0}{\hookrightarrow} t \\
&t \overset{\mathrm{k+1}}{\hookrightarrow} \begin{cases} \mathtt{fail} & \text{if } \mathsf{step}(t) = \mathtt{fail} \\ \mathtt{halt} & \text{if } \mathsf{step}(t) = \mathtt{halt} \\ t' & \text{if } \mathsf{step}(t) = t'' \text{ and } t'' \overset{\mathrm{k}}{\hookrightarrow} t' \end{cases}
\end{aligned}
$$

Define the set of possible statuses of a computation after some steps as $\mathcal{CS} = \{\mathtt{fail}, \mathtt{halt}, \mathtt{running}\}$. Define the set of possible endings of a computation as $\mathcal{CE} = \{\mathtt{fail}, \mathtt{halt}, \mathtt{diverge}\}$.

Define the function $\mathtt{observe\text{-}k}(\,\cdot\,) : \mathbb{N} \times \mathsf{Conf} \to \mathcal{CS}$, which tells whether a configuration can be observed for k steps, as follows:

$$\texttt{observe-k}(k,t) = \begin{cases} \texttt{fail} & \text{if } t \overset{k}{\hookrightarrow} \texttt{fail} \\ \texttt{halt} & \text{if } t \overset{k}{\hookrightarrow} \texttt{halt} \\ \texttt{running} & \text{if } \exists t'.t \overset{k}{\hookrightarrow} t' \end{cases}$$

Define the function $\texttt{observe}(\,\cdot\,) : \mathsf{Conf} \to \mathcal{CE}$, which tells the ending outcome of a configuration, as follows:

$$\texttt{observe}(t) = \begin{cases} \texttt{fail} & \text{if } \exists k \in \mathbb{N}.\texttt{observe-k}(k,t) = \texttt{fail} \\ \texttt{halt} & \text{if } \exists k \in \mathbb{N}.\texttt{observe-k}(k,t) = \texttt{halt} \\ \texttt{diverge} & \text{otherwise } (\forall k \in \mathbb{N}, \texttt{observe-k}(k,t) = \texttt{running}) \end{cases}$$

3.2 General World Specification

The general world specification is presented below.

$$\begin{aligned} \mathit{WSpec} \stackrel{\mathsf{def}}{=} \{&\mathsf{World}, \mathsf{lev}, \triangleright, \mathsf{O}, \sqsupseteq \ \mid \\ &\mathsf{World} \in \mathit{Set} && \wedge\, \mathsf{lev} \in \mathsf{World} \to \mathbb{N} \\ &\wedge \triangleright \in \mathsf{World} \to \mathsf{World} && \wedge\, \mathsf{O} \in \mathcal{P}(\mathcal{L}_1.\mathsf{Conf} \times \mathcal{L}_2.\mathsf{Conf}) \\ &\wedge \sqsupseteq \in \mathcal{P}(\mathsf{World} \times \mathsf{World}) && \wedge \sqsupseteq \text{ is a preorder} \\ &\wedge \forall \mathsf{W}' \sqsupseteq \mathsf{W}. \triangleright \mathsf{W}' \sqsupseteq \triangleright \mathsf{W} \\ &\wedge \forall \mathsf{W}. \triangleright \mathsf{W} \sqsupseteq \mathsf{W} && \wedge\, \forall \mathsf{W}' \sqsupseteq \mathsf{W}.\mathsf{lev}(\mathsf{W}') \leq \mathsf{lev}(\mathsf{W}) \\ &\wedge \forall \mathsf{W}.\mathsf{lev}(\mathsf{W}) > 0 \Rightarrow \mathsf{lev}(\triangleright \mathsf{W}) = \mathsf{lev}(\mathsf{W}) - 1\} \end{aligned}$$

A world specification must define what a world is (World), how many steps are left for the computation (lev, this is a trick needed for defining step-indexed logical relations that hide the step in the world), how to derive a 'later' world with smaller steps ($\triangleright$), how to observe configurations (O), how to define future worlds ($\sqsupseteq$) and public versions of future worlds ($\sqsupseteq$). This specification is given in general terms w.r.t. language specifications $\mathcal{L}_1$ and $\mathcal{L}_2$. It will be made concrete in Section 3.5 with instantiations of concrete language specifications $\mathit{LSpec}^{\lambda^{\tau}}$ and $\mathit{LSpec}^{\lambda^{u}}$ that are defined later on.

Define the strictly-future world relation, denoted with $\sqsupset$, as follows:

$$\sqsupset \stackrel{\mathsf{def}}{=} \{(\mathsf{W}', \mathsf{W}) \mid \mathsf{lev}(\mathsf{W}) > 0 \wedge \mathsf{W}' \sqsupseteq \triangleright \mathsf{W}\}$$

Use R to denote an arbitrary relation, i.e., a set of tuples of elements of set.
Define the set of world-value relations WVRel as follows: $\{R \in \mathcal{P}(\mathsf{World}, \mathcal{L}_1.\mathsf{Val}, \mathcal{L}_2.\mathsf{Val})\}$.
Define the values of a world-value relation R based on a world W as follows:

$$R(\mathsf{W}) = \{(v_1, v_2) \mid (\mathsf{W}, v_1, v_2) \in R\} \qquad \text{for } R \in \mathsf{WVRel}$$

Define the monotonic closure of a world-value relation R, denoted with $\square(\,\cdot\,)$, as follows:

$$\square(R) \stackrel{\mathsf{def}}{=} \{(\mathsf{W}, v_1, v_2) \mid \forall \mathsf{W}' \sqsupseteq \mathsf{W}.(\mathsf{W}', v_1, v_2) \in R\} \qquad \text{for } R \in \mathsf{WVRel}$$

Define the function for building of a world-value relation, denoted with $\mathsf{WVRel}(\cdot)$, as follows:

$$\mathsf{WVRel}(R_1, R_2) \stackrel{\mathsf{def}}{=} \{(\mathsf{W}, v_1, v_2) \mid \forall \mathsf{W}, v_1 \in R_1, v_2 \in R_2\}$$
$$\text{for } R_1 \subseteq \mathcal{L}_1.\mathsf{Val}, R_2 \subseteq \mathcal{L}_2.\mathsf{Val}$$

Note that function $\mathsf{WVRel}(\cdot)$, works on sets now, but it can be extended to work on relations as well.

Lemma 2 (Well-founded $\sqsupset$)**.** $\sqsupset$ *is well-founded.*

Proof. Because the level of the worlds strictly decrease. □

Lemma 3 (Properties of future worlds)**.**

$$\forall \underline{\mathsf{W}}, \underline{\mathsf{W}'}, \underline{\mathsf{W}''}.\underline{\mathsf{W}''} \sqsupset \underline{\mathsf{W}'} \textit{ and } \underline{\mathsf{W}'} \sqsupset \underline{\mathsf{W}} \textit{ then } \underline{\mathsf{W}''} \sqsupset \underline{\mathsf{W}}$$
$$\forall \underline{\mathsf{W}}, \underline{\mathsf{W}'}, \underline{\mathsf{W}''}.\underline{\mathsf{W}''} \sqsupset \underline{\mathsf{W}'} \textit{ and } \underline{\mathsf{W}'} \sqsupseteq \underline{\mathsf{W}} \textit{ then } \underline{\mathsf{W}''} \sqsupset \underline{\mathsf{W}}$$
$$\forall \underline{\mathsf{W}}, \underline{\mathsf{W}'}, \underline{\mathsf{W}''}.\underline{\mathsf{W}''} \sqsupseteq \underline{\mathsf{W}'} \textit{ and } \underline{\mathsf{W}'} \sqsupset \underline{\mathsf{W}} \textit{ then } \underline{\mathsf{W}''} \sqsupset \underline{\mathsf{W}}$$

Proof. By definition of $\sqsupset$ and $\triangleright$, lev, $\sqsupseteq$. □

3.3 Language Specification for λ^{τ}

$\mathit{LSpec}^{\lambda^{\tau}}$ is the language specification for λ^{τ}.

$$\mathbf{Val} \stackrel{\mathsf{def}}{=} \{\mathbf{v}\} \qquad \mathbf{Ter} \stackrel{\mathsf{def}}{=} \{\mathbf{t}\}$$

$$\mathbf{Con} \stackrel{\mathsf{def}}{=} \{\mathbb{E}\} \qquad \mathbf{Conf} \stackrel{\mathsf{def}}{=} \{\mathbf{t}\}$$

$$\mathbf{plugv}(\mathbf{v}, \mathbb{E}) \stackrel{\mathsf{def}}{=} \mathbb{E}[\mathbf{v}] \qquad \mathbf{plugc}(\mathbf{t}, \mathbb{E}) \stackrel{\mathsf{def}}{=} \mathbb{E}[\mathbf{t}]$$

$$\mathbf{step}(\mathbf{t}) \stackrel{\mathsf{def}}{=} \begin{cases} \mathbf{t}' & \text{if } \mathbf{t} \hookrightarrow \mathbf{t}' \\ \mathtt{halt} & \text{if } \mathbf{t} \in \mathbf{Val} \\ \mathtt{fail} & \text{otherwise} \end{cases} \qquad \mathbf{oftype}(\tau) \stackrel{\mathsf{def}}{=} \{\mathbf{v} \mid \emptyset \vdash \mathbf{v} : \tau\}$$

$$\mathbf{unit}(\mathbf{v}) \stackrel{\mathsf{def}}{=} \{\mathtt{unit}\}$$

$$\mathbf{bool}(\mathbf{v}) \stackrel{\mathsf{def}}{=} \{\mathtt{true}, \mathtt{false}\} \qquad \mathbf{pair}(\mathbf{v_1}, \mathbf{v_2}) \stackrel{\mathsf{def}}{=} \{\langle \mathbf{v_1}, \mathbf{v_2} \rangle\}$$

$$\mathbf{appl}(\mathbf{v_1}, \mathbf{v_2}) \stackrel{\mathsf{def}}{=} \{\mathbf{t} \in \mathbf{Ter} \mid \exists \mathbf{t}', \mathbf{x}, \tau. \mathbf{v_1} \equiv \lambda \mathbf{x} : \tau. \mathbf{t}' \wedge \mathbf{t} \equiv \mathbf{t}'[\mathbf{x} \mapsto \mathbf{v_2}]\}$$

$$\mathbf{inl}(\mathbf{v}) \stackrel{\mathsf{def}}{=} \{\mathbf{v}' \mid \mathbf{v}' \equiv \mathrm{inl}\ \mathbf{v}\} \qquad \mathbf{inr}(\mathbf{v}) \stackrel{\mathsf{def}}{=} \{\mathbf{v}' \mid \mathbf{v}' \equiv \mathrm{inr}\ \mathbf{v}\}$$

$$\begin{aligned} \mathit{LSpec}^{\lambda^{\tau}} \stackrel{\mathsf{def}}{=} (&\mathbf{Val}, \mathbf{Ter}, \mathbf{Con}, \mathbf{Conf}, \mathbf{plugv}(\cdot), \mathbf{plugc}(\cdot), \mathbf{step}(\cdot), \\ &\mathbf{oftype}(\cdot), \mathbf{unit}(\cdot), \mathbf{bool}(\cdot), \mathbf{pair}(\cdot), \mathbf{appl}(\cdot), \mathbf{inl}(\mathbf{v}), \mathbf{inr}(\mathbf{v})) \end{aligned}$$

To ensure this definition is correct, $\mathit{LSpec}^{\lambda^{\tau}}$ must be included in the general language specification LSpec (Theorem 1).

Theorem 1 (Correctness of $\mathit{LSpec}^{\lambda^{\tau}}$)**.** $\mathit{LSpec}^{\lambda^{\tau}} \in \mathit{LSpec}$

Proof of Theorem 1. Trivial. □

3.4 Language Specification for λ^{u}

$\mathit{LSpec}^{\lambda^{\mathsf{u}}}$ is the language specification for λ^{u}.

$$\mathsf{Val} \stackrel{\mathsf{def}}{=} \{\mathsf{v}\} \qquad \mathsf{Ter} \stackrel{\mathsf{def}}{=} \{\mathsf{t}\}$$

$$\mathsf{Con} \stackrel{\mathsf{def}}{=} \{\mathbb{E}\} \qquad \mathsf{Conf} \stackrel{\mathsf{def}}{=} \{\mathsf{t}\}$$

$$\mathsf{plugv}(\mathsf{v}, \mathbb{E}) \stackrel{\mathsf{def}}{=} \mathbb{E}[\mathsf{v}] \qquad \mathsf{plugc}(\mathsf{t}, \mathbb{E}) \stackrel{\mathsf{def}}{=} \mathbb{E}[\mathsf{t}]$$

$$\mathsf{step}(\mathsf{t}) \stackrel{\mathsf{def}}{=} \begin{cases} \mathsf{t}' & \text{if } \mathsf{t} \hookrightarrow \mathsf{t}' \\ \mathtt{halt} & \text{if } \mathsf{t} \in \mathsf{Val} \\ \mathtt{fail} & \text{otherwise} \end{cases}$$

$$\mathsf{oftype}(\tau) \stackrel{\mathsf{def}}{=} \left\{ \mathsf{v} \;\middle|\; \begin{array}{ll} \exists \mathsf{t}'.\mathsf{v} \equiv \lambda \mathsf{x}.\ \mathsf{t}' & \text{if } \tau \equiv \tau_1 \to \tau_2 \\ \begin{array}{l}\exists \mathsf{v}_1, \mathsf{v}_2.\mathsf{v} \equiv \langle \mathsf{v}_1, \mathsf{v}_2 \rangle \wedge \\ \quad \mathsf{v}_1 \in \mathsf{oftype}(\tau_1) \wedge \mathsf{v}_2 \in \mathsf{oftype}(\tau_2)\end{array} & \text{if } \tau \equiv \tau_1 \times \tau_2 \\ \begin{array}{l}\exists \mathsf{v}_1.\mathsf{v} \equiv \mathrm{inl}\ \mathsf{v}_1 \wedge \mathsf{v}_1 \in \mathsf{oftype}(\tau_1) \text{ or} \\ \qquad \exists \mathsf{v}_2.\mathsf{v} \equiv \mathrm{inr}\ \mathsf{v}_2 \wedge \mathsf{v}_2 \in \mathsf{oftype}(\tau_2)\end{array} & \text{if } \tau \equiv \tau_1 \uplus \tau_2 \\ \mathsf{v} \equiv \mathtt{unit} & \text{if } \tau \equiv \mathtt{Unit} \\ \begin{array}{l}\mathsf{v} \equiv \mathtt{true} \vee \\ \quad \mathsf{v} \equiv \mathtt{false}\end{array} & \text{if } \tau \equiv \mathtt{Bool} \end{array} \right\}$$

$$\mathsf{unit}(\mathsf{v}) \stackrel{\mathsf{def}}{=} \{\mathtt{unit}\}$$

$$\mathsf{bool}(\mathsf{v}) \stackrel{\mathsf{def}}{=} \{\mathtt{true}, \mathtt{false}\}$$

$$\mathsf{pair}(\mathsf{v}_1, \mathsf{v}_2) \stackrel{\mathsf{def}}{=} \{\langle \mathsf{v}_1, \mathsf{v}_2 \rangle\}$$

$$\mathsf{appl}(\mathsf{v}_1, \mathsf{v}_2) \stackrel{\mathsf{def}}{=} \{\mathsf{t} \in \mathsf{Ter} \mid \exists \mathsf{t}', \mathsf{x}.\mathsf{v}_1 \equiv \lambda \mathsf{x}.\mathsf{t}' \wedge \mathsf{t} \equiv \mathsf{t}'[\mathsf{x} \mapsto \mathsf{v}_2]\}$$

$$\mathsf{inl}(\mathsf{v}) \stackrel{\mathsf{def}}{=} \{\mathsf{v}' \mid \mathsf{v}' \equiv \mathrm{inl}\ \mathsf{v}\}$$

$$\mathsf{inr}(\mathsf{v}) \stackrel{\mathsf{def}}{=} \{\mathsf{v}' \mid \mathsf{v}' \equiv \mathrm{inr}\ \mathsf{v}\}$$

$$\begin{aligned} \mathit{LSpec}^{\lambda^{\mathsf{u}}} \stackrel{\mathsf{def}}{=} (&\mathsf{Val}, \mathsf{Ter}, \mathsf{Con}, \mathsf{Conf}, \mathsf{plugv}(\cdot), \mathsf{plugc}(\cdot), \mathsf{step}(\cdot), \\ &\mathsf{oftype}(\cdot), \mathsf{unit}(\cdot), \mathsf{bool}(\cdot), \mathsf{pair}(\cdot), \mathsf{appl}(\cdot), \mathsf{inl}(\mathsf{v}), \mathsf{inr}(\mathsf{v})) \end{aligned}$$

To ensure this definition is correct, $\mathit{LSpec}^{\lambda^{\mathsf{u}}}$ must be included in the general language specification LSpec (Theorem 2).

Theorem 2 (Correctness of $\mathit{LSpec}^{\lambda^{\mathsf{u}}}$)**.** $\mathit{LSpec}^{\lambda^{\mathsf{u}}} \in \mathit{LSpec}$

Proof of Theorem 2. Trivial. □

3.5 World Specification

This Section presents $\underline{\mathcal{W}}$, a concrete instantiation of the *WSpec* of Section 3.2 to be used by the logical relation between concrete language specifications.

$$
\begin{aligned}
\mathsf{World}^{\underline{\mathcal{W}}} &\stackrel{\mathsf{def}}{=} \{\underline{\mathsf{W}} = (k) \mid k \in \mathbb{N}\} \\
\mathsf{lev}(\underline{\mathsf{W}}) &\stackrel{\mathsf{def}}{=} \underline{\mathsf{W}}.k \\
\triangleright(0) &\stackrel{\mathsf{def}}{=} (0) \\
\triangleright(k+1) &\stackrel{\mathsf{def}}{=} (k) \\
\mathsf{O}(\underline{\mathsf{W}})_{\lesssim} &\stackrel{\mathsf{def}}{=} \left\{(\mathbf{t}, \mathsf{t}) \;\middle|\; \begin{array}{r} (LSpec^{\lambda^{\tau}}.\mathtt{observe\text{-}k}(\mathsf{lev}(\underline{\mathsf{W}}), \mathbf{t}) = \mathtt{halt} \Rightarrow \\ \exists k.\, LSpec^{\lambda^{u}}.\mathtt{observe\text{-}k}(k, \mathsf{t}) = \mathtt{halt}) \end{array}\right\} \\
\mathsf{O}(\underline{\mathsf{W}})_{\gtrsim} &\stackrel{\mathsf{def}}{=} \left\{(\mathbf{t}, \mathsf{t}) \;\middle|\; \begin{array}{r} (LSpec^{\lambda^{u}}.\mathtt{observe\text{-}k}(\mathsf{lev}(\underline{\mathsf{W}}), \mathsf{t}) = \mathtt{halt} \Rightarrow \\ \exists k.\, LSpec^{\lambda^{\tau}}.\mathtt{observe\text{-}k}(k, \mathbf{t}) = \mathtt{halt}) \end{array}\right\} \\
\mathsf{O}(\underline{\mathsf{W}})_{\approx} &\stackrel{\mathsf{def}}{=} \mathsf{O}(\underline{\mathsf{W}})_{\lesssim} \cap \mathsf{O}(\underline{\mathsf{W}})_{\gtrsim} \\
(k) \sqsupseteq (k') &\stackrel{\mathsf{def}}{=} k \leq k' \\
\underline{\mathcal{W}} &\in \{\mathsf{World}^{\underline{\mathcal{W}}}, \mathsf{lev}^{\underline{\mathcal{W}}}, \triangleright, \mathsf{O}^{\underline{\mathcal{W}}}, \sqsupseteq\}
\end{aligned}
$$

To ensure this definition is correct, $\underline{\mathcal{W}}$ must be included in the general language specification *WSpec* (Theorem 3).

Theorem 3 (Correctness of $\underline{\mathcal{W}}$)**.** *$\underline{\mathcal{W}} \in WSpec$*

Proof of Theorem 3. Trivial. $\square$

In subsequent sections, we will regularly use $\lesssim$, $\gtrsim$ and $\approx$ as subscripts on logical relations and so on, to indicate that they should be interpreted w.r.t. the worldspec with the corresponding $\mathsf{O}(\underline{\mathsf{W}})$. We will use $\square$ as a meta-variable that can be instantiated to either $\lesssim$,$\gtrsim$, or $\approx$ for those theorems or definitions that work for all three.

Lemma 4 (Observation relation closed under antireduction)**.** *If $\mathbf{t} \hookrightarrow^{\mathbf{i}} \mathbf{t}'$ and $\mathsf{t} \hookrightarrow^{\mathsf{j}} \mathsf{t}'$, $(\mathbf{t}', \mathsf{t}') \in \mathsf{O}(\underline{\mathsf{W}}')_{\square}$, $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $\mathsf{lev}(\underline{\mathsf{W}}') \geq \mathsf{lev}(\underline{\mathsf{W}}) - \min(i, j)$, i.e. $\mathsf{lev}(\underline{\mathsf{W}}) \leq \mathsf{lev}(\underline{\mathsf{W}}') + \min(i, j)$, then $(\mathbf{t}, \mathsf{t}) \in \mathsf{O}(\underline{\mathsf{W}})_{\square}$.*

Proof. If $\mathbf{t}'$ and t' halt, then so do $\mathbf{t}$ and t. Otherwise, if $\mathbf{t}'$ and t' take at least $\mathsf{lev}(\underline{\mathsf{W}}')$ steps, then $\mathbf{t}$ and t take at least $\mathsf{lev}(\underline{\mathsf{W}}') + \mathtt{min}(i, j)$ steps. $\square$

Lemma 5 (No observation with 0 steps)**.** *If $\mathsf{lev}(\underline{\mathsf{W}}) = 0$, then for all $\mathbf{t}$, t, we have that $(\mathbf{t}, \mathsf{t}) \in \mathsf{O}(\underline{\mathsf{W}})_{\square}$.*

Proof. Just a bit of definition unfolding. $\square$

Lemma 6 (Source divergence is target divergence or failure)**.** *If $\mathbf{t} \Uparrow$ and either $\mathsf{t} \Uparrow$ or $\mathsf{t} \hookrightarrow^{*} \mathsf{wrong}$, i.e. $\mathbf{t}$ diverges and t either diverges or fails, then we have that $(\mathbf{t}, \mathsf{t}) \in \mathsf{O}(\underline{\mathsf{W}})_{\square}$.*

Proof. Just a bit of definition unfolding. □

Lemma 7 (No steps means relation)**.** *If* $LSpec^{\lambda^{\tau}}$.observe-k$(\mathsf{lev}(\underline{\mathsf{W}}), \mathbf{t}) = \texttt{running}$ *and* $LSpec^{\lambda^{u}}$.observe-k$(\mathsf{lev}(\underline{\mathsf{W}}), \mathsf{t}) = \texttt{running}$*, i.e. both* $\mathbf{t}$ *and* t *run out of steps in world* $\underline{\mathsf{W}}$*, then we have that* $(\mathbf{t}, \mathsf{t}) \in \mathsf{O}(\underline{\mathsf{W}})_{\square}$*.*

Proof. Just a bit of definition unfolding. □

4 Logical Relations

This Section defines the logical relations used to prove properties of the compiler. Instead of giving general logical relations as Hur and Dreyer, a specific logical relations is given, between source and target language specifications.

The logical relations between $LSpec^{\lambda^{\tau}}$ and $LSpec^{\lambda^{u}}$ are defined based on a relation on values $\mathcal{V}[\![\cdot]\!]_{\square}$, continuations $\mathcal{K}[\![\cdot]\!]_{\square}$, terms (also called computations) $\mathcal{E}[\![\cdot]\!]_{\square}$ and based on an interpretation for typing environments $\mathcal{G}[\![\cdot]\!]_{\square}$. These logical relations are used to relate $LSpec^{\lambda^{u}}$ and $LSpec^{\lambda^{\tau}}$, so their definition contains terms of the two language specifications in place of elements of abstract language specifications and elements of $\underline{\mathcal{W}}$ in place of elements of an abstract world specification.

Pseudo-type $\hat{\tau}$.

$$\hat{\tau} ::= \texttt{Bool} \mid \texttt{Unit} \mid \hat{\tau}\times\hat{\tau} \mid \hat{\tau}\uplus\hat{\tau} \mid \hat{\tau}\to\hat{\tau} \mid \texttt{EmulDV}_{n;p}$$
$$\hat{\mathbf{\Gamma}} ::= \emptyset \mid \hat{\mathbf{\Gamma}}, \mathbf{x} : \hat{\tau}$$

Helper functions for $\texttt{EmulDV}$.

$$\texttt{toEmul}(\emptyset)_{n;p} = \emptyset \qquad \texttt{toEmul}(\Gamma, x)_{n;p} = \texttt{toEmul}(\Gamma)_{n;p}, (\mathbf{x} : \texttt{EmulDV}_{n;p})$$
$$\texttt{repEmul}(\emptyset) = \emptyset \qquad \texttt{repEmul}(\mathbf{\Gamma}, (\mathbf{x} : \hat{\tau})) = \texttt{repEmul}(\mathbf{\Gamma}), (\mathbf{x} : \texttt{repEmul}(\hat{\tau}))$$

$$\begin{aligned}
\texttt{repEmul}(\hat{\tau}\times\hat{\tau}') &= \texttt{repEmul}(\hat{\tau})\times\texttt{repEmul}(\hat{\tau}')\\
\texttt{repEmul}(\hat{\tau}\uplus\hat{\tau}') &= \texttt{repEmul}(\hat{\tau})\uplus\texttt{repEmul}(\hat{\tau}')\\
\texttt{repEmul}(\hat{\tau}\to\hat{\tau}') &= \texttt{repEmul}(\hat{\tau})\to\texttt{repEmul}(\hat{\tau}')\\
\texttt{repEmul}(\texttt{EmulDV}_{n;p}) &= \mathsf{UVal}_{n}\\
\texttt{repEmul}(\texttt{Bool}) &= \texttt{Bool}\\
\texttt{repEmul}(\texttt{Unit}) &= \texttt{Unit}
\end{aligned}$$

$\texttt{oftype}(\cdot)$ definition.

$$\mathbf{oftype}(\hat{\tau}) \stackrel{\mathsf{def}}{=} \{\mathbf{v} \mid \emptyset \vdash \mathbf{v} : \texttt{repEmul}(\hat{\tau})\}$$

$$\mathsf{oftype}(\hat{\tau}) \stackrel{\mathsf{def}}{=} \left\{ \mathsf{v} \;\middle|\; \begin{array}{ll}
\mathsf{v} = \texttt{unit} & \text{if } \hat{\tau} = \texttt{Unit}\\
\mathsf{v} = \texttt{true} \text{ or } \mathsf{v} = \texttt{false} & \text{if } \hat{\tau} = \texttt{Bool}\\
\exists \mathsf{t}.\, \mathsf{v} = \lambda \mathsf{x}.\, \mathsf{t} & \text{if } \exists \hat{\tau_1}, \hat{\tau_2}.\, \hat{\tau} = \hat{\tau_1}\to\hat{\tau_2}\\
\exists \mathsf{v}_1 \in \mathsf{oftype}(\hat{\tau_1}), \mathsf{v}_2 \in \mathsf{oftype}(\hat{\tau_2}).\, \mathsf{v} = \langle \mathsf{v}_1, \mathsf{v}_2\rangle & \text{if } \exists \hat{\tau_1}, \hat{\tau_2}.\, \hat{\tau} = \hat{\tau_1}\times\hat{\tau_2}\\
\exists \mathsf{v}_1 \in \mathsf{oftype}(\hat{\tau_1}).\, \mathsf{v} = \mathrm{inl}\ \mathsf{v}_1 \text{ or } \exists \mathsf{v}_2 \in \mathsf{oftype}(\hat{\tau_2}).\, \mathsf{v} = \mathrm{inr}\ \mathsf{v}_2 & \text{if } \exists \hat{\tau_1}, \hat{\tau_2}.\, \hat{\tau} = \hat{\tau_1}\uplus\hat{\tau_2}
\end{array} \right\}$$

$$\texttt{oftype}(\hat{\tau}) \stackrel{\mathsf{def}}{=} \{(\mathbf{v}, \mathsf{v}) \mid \mathbf{v} \in \mathbf{oftype}(\hat{\tau}) \text{ and } \mathsf{v} \in \mathsf{oftype}(\hat{\tau})\}$$

Logical relations for values ($\mathcal{V}[\![\cdot]\!]_{\square}$), contexts ($\mathcal{K}[\![\cdot]\!]_{\square}$), terms ($\mathcal{E}[\![\cdot]\!]_{\square}$) and environments ($\mathcal{G}[\![\cdot]\!]_\Box$).

$$\triangleright R \stackrel{\mathsf{def}}{=} \{(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \mid \mathsf{lev}(\underline{\mathsf{W}}) > 0 \Rightarrow (\triangleright \underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in R\}$$

$$\mathcal{V}[\![\mathtt{Unit}]\!]_\Box \stackrel{\mathsf{def}}{=} \{(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \mid (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \Box(\mathsf{WVRel}(\mathbf{unit}(\mathtt{unit}), \mathsf{unit}(\mathtt{unit})))\}$$

$$\mathcal{V}[\![\mathtt{Bool}]\!]_\Box \stackrel{\mathsf{def}}{=} \{(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \mid \exists v \in [\![\mathtt{Bool}]\!].\, (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \Box(\mathsf{WVRel}(\mathbf{bool}(\mathbf{v}), \mathsf{bool}(\mathsf{v})))\}$$

$$\mathcal{V}[\![\hat{\tau}' \to \hat{\tau}]\!]_\Box \stackrel{\mathsf{def}}{=} \left\{ (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \;\middle|\; \begin{array}{c} (\mathbf{v}, \mathsf{v}) \in \mathtt{oftype}(\hat{\tau}' \to \hat{\tau}) \text{ and} \\ \forall (\underline{\mathsf{W}}', \underline{\mathsf{W}}) \in \sqsupset, \forall (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\hat{\tau}']\!]_\Box, \\ \forall \mathbf{t} \in \mathbf{appl}(\mathbf{v}, \mathbf{v}'), \\ \forall \mathsf{t} \in \mathsf{appl}(\mathsf{v}, \mathsf{v}'), (\underline{\mathsf{W}}', \mathbf{t}, \mathsf{t}) \in \mathcal{E}[\![\hat{\tau}]\!]_\Box \end{array} \right\}$$

$$\mathcal{V}[\![\hat{\tau_1} \times \hat{\tau_2}]\!]_\Box \stackrel{\mathsf{def}}{=} \left\{ (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \;\middle|\; \begin{array}{c} (\mathbf{v}, \mathsf{v}) \in \mathtt{oftype}(\hat{\tau_1} \times \hat{\tau_2}) \text{ and} \\ \exists (\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \triangleright \mathcal{V}[\![\hat{\tau_1}]\!]_\Box, \\ \exists (\underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \triangleright \mathcal{V}[\![\hat{\tau_2}]\!]_\Box, \\ (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \Box(\mathsf{WVRel}(\mathbf{pair}(\mathbf{v_1}, \mathbf{v_2}), \mathsf{pair}(\mathsf{v}_1, \mathsf{v}_2))) \end{array} \right\}$$

$$\mathcal{V}[\![\hat{\tau_1} \uplus \hat{\tau_2}]\!]_\Box \stackrel{\mathsf{def}}{=} \left\{ (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \;\middle|\; \begin{array}{l} (\mathbf{v}, \mathsf{v}) \in \mathtt{oftype}(\hat{\tau_1} \times \hat{\tau_2}) \text{ and} \\ \exists \mathbf{v}', \mathsf{v}'.\ ((\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \triangleright \mathcal{V}[\![\hat{\tau_1}]\!]_\Box \text{ and} \\ (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \Box(\mathsf{WVRel}(\mathbf{inl}(\mathbf{v}'), \mathsf{inl}(\mathsf{v}')))) \text{ or} \\ \exists \mathbf{v}', \mathsf{v}'.\ ((\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \triangleright \mathcal{V}[\![\hat{\tau_2}]\!]_\Box \text{ and} \\ (\mathbf{v}, \mathsf{v}) \in \Box(\mathsf{WVRel}(\underline{\mathsf{W}}, \mathbf{inr}(\mathbf{v}'), \mathsf{inr}(\mathsf{v}')))) \end{array} \right\}$$

$$\mathcal{V}[\![\mathtt{EmulDV}_{0;p}]\!]_\Box \stackrel{\mathsf{def}}{=} \{(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \mid \mathbf{v} = \mathtt{unit} \text{ and } p = \mathtt{imprecise}\}$$

$$\mathcal{V}[\![\mathtt{EmulDV}_{n+1;p}]\!]_\Box \stackrel{\mathsf{def}}{=} \left\{ (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \;\middle|\; \begin{array}{l} \mathbf{v} \in \mathbf{oftype}(\mathrm{UVal}_{n+1}) \text{ and one of the following holds:} \\ \quad \left\{ \begin{array}{l} \mathbf{v} = \mathbf{in}_{\mathsf{unk};\mathbf{n}} \text{ and } p = \mathtt{imprecise} \\ \exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ \mathbf{v}' \text{ and } (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{Unit}]\!]_\Box \\ \exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ \mathbf{v}' \text{ and } (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{Bool}]\!]_\Box \\ \exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\times;\mathbf{n}}\ \mathbf{v}' \text{ and} \\ \quad (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \times \mathtt{EmulDV}_{n;p}]\!]_\Box \\ \exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}\ \mathbf{v}' \text{ and} \\ \quad (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \uplus \mathtt{EmulDV}_{n;p}]\!]_\Box \\ \exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\to;\mathbf{n}}\ \mathbf{v}' \text{ and} \\ \quad (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \to \mathtt{EmulDV}_{n;p}]\!]_\Box \end{array} \right. \end{array} \right\}$$

$$\begin{aligned} \mathcal{K}[\![\hat{\tau}]\!]_\Box &\stackrel{\mathsf{def}}{=} \{(\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \mid \forall \underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}, \forall (\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\hat{\tau}]\!]_\Box, \forall \mathbf{t} \in \mathbf{plugv}(\mathbf{v}, \mathbb{E}), \\ &\quad \forall \mathsf{t} \in \mathsf{plugv}(\mathsf{v}, \mathbb{E}), (\mathbf{t}, \mathsf{t}) \in \mathsf{O}(\underline{\mathsf{W}}')\} \\ \mathcal{E}[\![\hat{\tau}]\!]_\Box &\stackrel{\mathsf{def}}{=} \{(\underline{\mathsf{W}}, \mathbf{t}, \mathsf{t}) \mid \forall (\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\hat{\tau}]\!]_\Box, \forall \mathbf{t}' \in \mathbf{plugc}(\mathbf{t}, \mathbb{E}), \\ &\quad \forall \mathsf{t}' \in \mathsf{plugc}(\mathsf{t}, \mathbb{E}), (\mathbf{t}', \mathsf{t}') \in \mathsf{O}(\underline{\mathsf{W}})\} \\ \mathcal{G}[\![\emptyset]\!]_\Box &\stackrel{\mathsf{def}}{=} \{(\underline{\mathsf{W}}, \emptyset, \emptyset)\} \\ \mathcal{G}[\![\hat{\mathbf{\Gamma}}, (\mathbf{x} : \hat{\tau})]\!]_\Box &\stackrel{\mathsf{def}}{=} \{(\underline{\mathsf{W}}, \gamma[\mathbf{x} \mapsto \mathbf{v}], \gamma[\mathsf{x} \mapsto \mathsf{v}]) \mid (\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\hat{\mathbf{\Gamma}}]\!]_\Box \text{ and } (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\hat{\tau}]\!]_\Box\} \end{aligned}$$

Define relatedness of open terms when closing them with related substitutions produces closed terms that are related by the expression relation.

Definition 4 (Logical relation up to n steps)**.**

$$\begin{aligned}\hat{\boldsymbol{\Gamma}} \vdash \mathbf{t} \,\square_{\mathsf{n}}\, \mathsf{t} : \hat{\tau} \stackrel{\mathsf{def}}{=} & \ \mathtt{repEmul}(\hat{\boldsymbol{\Gamma}}) \vdash \mathbf{t} : \mathtt{repEmul}(\hat{\tau}) \ \textit{and}\ \mathtt{dom}(\hat{\boldsymbol{\Gamma}}) \vdash \mathsf{t} \\ & \textit{and}\ \forall \underline{\mathsf{W}}.\, \mathsf{lev}(\underline{\mathsf{W}}) \leq n \Rightarrow \forall (\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\hat{\boldsymbol{\Gamma}}]\!]_{\square}.\, (\underline{\mathsf{W}}, \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}\end{aligned}$$

Definition 5 (Logical relation)**.**

$$\hat{\boldsymbol{\Gamma}} \vdash \mathbf{t} \,\square\, \mathsf{t} : \hat{\tau} \stackrel{\mathsf{def}}{=} \hat{\boldsymbol{\Gamma}} \vdash \mathbf{t} \,\square_{\mathsf{n}}\, \mathsf{t} : \hat{\tau} \ \textit{for all } n$$

We also define a logical relation for program contexts:

Definition 6 (Logical relation for contexts)**.**

$$\begin{aligned}\vdash \mathfrak{C} \,\square\, \mathfrak{C} : \hat{\boldsymbol{\Gamma}}', \hat{\tau}' \to \hat{\boldsymbol{\Gamma}}, \hat{\tau} \stackrel{\mathsf{def}}{=} & \\ & \vdash \mathfrak{C} : \mathtt{repEmul}(\hat{\boldsymbol{\Gamma}}'), \mathtt{repEmul}(\hat{\tau}') \to \mathtt{repEmul}(\hat{\boldsymbol{\Gamma}}), \mathtt{repEmul}(\hat{\tau}) \\ & \textit{and}\ \vdash \mathfrak{C} : \mathtt{dom}(\hat{\boldsymbol{\Gamma}}') \to \mathtt{dom}(\hat{\boldsymbol{\Gamma}}) \\ & \textit{and for all}\ \mathbf{t}, \mathsf{t}.\ \textit{if}\ \hat{\boldsymbol{\Gamma}}' \vdash \mathbf{t} \,\square\, \mathsf{t} : \hat{\tau}', \\ & \textit{then}\ \hat{\boldsymbol{\Gamma}} \vdash \mathfrak{C}[\mathbf{t}] \,\square\, \mathfrak{C}[\mathsf{t}] : \hat{\tau}\end{aligned}$$

Lemma 8 (Closedness under antireduction)**.** *If* $\mathbb{E}[\mathbf{t}] \hookrightarrow^{\mathsf{i}} \mathbb{E}[\mathbf{t}']$ *and* $\mathbb{E}[\mathsf{t}] \hookrightarrow^{\mathsf{j}} \mathbb{E}[\mathsf{t}']$ *for any* $\mathbb{E}$ *and* $\mathbb{E}$*,* $(\underline{\mathsf{W}}', \mathbf{t}', \mathsf{t}') \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}$*,* $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$*,* $\mathsf{lev}(\underline{\mathsf{W}}') \geq \mathsf{lev}(\underline{\mathsf{W}}) - \min(i,j)$*, i.e.* $\mathsf{lev}(\underline{\mathsf{W}}) \leq \mathsf{lev}(\underline{\mathsf{W}}') + \min(i,j)$*, then* $(\underline{\mathsf{W}}, \mathbf{t}, \mathsf{t}) \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}$.

Proof. Take an arbitrary $(\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$. Then we need to prove that $(\mathbb{E}[\mathbf{t}], \mathbb{E}[\mathsf{t}]) \in \mathsf{O}(\underline{\mathsf{W}})$. By Lemma 4, it suffices to prove that $(\mathbb{E}[\mathbf{t}'], \mathbb{E}[\mathsf{t}']) \in \mathsf{O}(\underline{\mathsf{W}}')$. By Lemma 12, we have that $(\underline{\mathsf{W}}', \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$, so that the result follows from $(\underline{\mathsf{W}}', \mathbf{t}', \mathsf{t}') \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}$. ◻

Lemma 9 (Later operator preserves monotonicity)**.** $\forall R, R \subseteq \square(R) \Rightarrow \triangleright R \subseteq \square(\triangleright R)$

Proof. By definition and assumptions on $\triangleright$ and lev. ◻

Lemma 10 (Term relations include value relations)**.** $\forall \hat{\tau}, \mathcal{V}[\![\hat{\tau}]\!]_{\square} \subseteq \mathcal{E}[\![\hat{\tau}]\!]_{\square}$.

Proof. Simple induction on $\hat{\tau}$. ◻

Lemma 11 (Monotonicity for environment relation)**.** *If* $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$*, then* $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\boldsymbol{\Gamma}]\!]_{\square}$ *implies that* $(\underline{\mathsf{W}}', \gamma, \gamma) \in \mathcal{G}[\![\boldsymbol{\Gamma}]\!]_{\square}$.

Proof. By definition. ◻

Lemma 12 (Monotonicity for continuation relation)**.** *If* $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$*, then* $(\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$ *implies that* $(\underline{\mathsf{W}}', \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$.

Proof. By definition. ◻

Lemma 13 (Monotonicity for value relation)**.** $\mathcal{V}[\![\hat{\tau}]\!]_{\square} \subseteq \square(\mathcal{V}[\![\hat{\tau}]\!])_{\square}$

Proof. By induction on $\hat{\tau}$. Definitions for all cases are monotone. The inductive cases follow by Lemma 9 and Lemma 3. $\square$

Lemma 14 (Adequacy for $\lesssim$)**.** *If* $\emptyset \vdash \mathbf{t} \lesssim_n \mathsf{t} : \tau$, *and if* $\mathbf{t} \hookrightarrow^{\mathbf{m}} \mathbf{v}$ *with* $n \geq m$, *then also* $\mathsf{t}\Downarrow$.

Proof. We have directly that $(\underline{\mathsf{W}}, \mathbf{t}, \mathsf{t}) \in \mathcal{E}[\![\tau]\!]_{\lesssim}$ for a world $\underline{\mathsf{W}}$ such that $\mathsf{lev}(\underline{\mathsf{W}}) = n$. Since $(\underline{\mathsf{W}}, \cdot, \cdot) \in \mathcal{K}[\![\tau]\!]_{\lesssim}$, we have that $(\mathbf{t}, \mathsf{t}) \in \mathsf{O}(\underline{\mathsf{W}})_{\lesssim}$. Since $LSpec^{\lambda^{\tau}}$.`observe-k`$(\mathsf{lev}(\underline{\mathsf{W}}), \mathbf{t}) =$ `halt`, we have by definition of $\mathsf{O}(\underline{\mathsf{W}})_{\lesssim}$ that $LSpec^{\lambda^{u}}$.`observe-k`$(k, \mathsf{t}) =$ `halt` for some k, i.e. $\mathsf{t}\Downarrow$. $\square$

Lemma 15 (Adequacy for $\gtrsim$)**.** *If* $\emptyset \vdash \mathbf{t} \gtrsim_n \mathsf{t} : \tau$ *and if* $\mathsf{t} \hookrightarrow^{\mathsf{m}} \mathsf{v}$ *with* $n \geq m$, *then also* $\mathbf{t}\Downarrow$.

Proof. We have directly that $(\underline{\mathsf{W}}, \mathbf{t}, \mathsf{t}) \in \mathcal{E}[\![\tau]\!]_{\gtrsim}$ for a world $\underline{\mathsf{W}}$ such that $\mathsf{lev}(\underline{\mathsf{W}}) = n$. Since $(\underline{\mathsf{W}}, \cdot, \cdot) \in \mathcal{K}[\![\tau]\!]_{\gtrsim}$, we have that $(\mathbf{t}, \mathsf{t}) \in \mathsf{O}(\underline{\mathsf{W}})_{\gtrsim}$. Since $LSpec^{\lambda^{u}}$.`observe-k`$(\mathsf{lev}(\underline{\mathsf{W}}), \mathsf{t}) =$ `halt`, we have by definition of $\mathsf{O}(\underline{\mathsf{W}})_{\gtrsim}$ that $LSpec^{\lambda^{\tau}}$.`observe-k`$(k, \mathbf{t}) =$ `halt` for some k, i.e. $\mathbf{t}\Downarrow$. $\square$

Lemma 16 (Adequacy for $\lesssim$ and $\gtrsim$)**.** *If* $\emptyset \vdash \mathbf{t} \lesssim_{\mathsf{n}} \mathsf{t} : \tau$, *and if* $\mathbf{t} \hookrightarrow^{m} \mathbf{v}$ *with* $n \geq m$, *then also* $\mathsf{t}\Downarrow$.

If $\emptyset \vdash \mathbf{t} \gtrsim_{\mathsf{n}} \mathsf{t} : \tau$ *and if* $\mathsf{t} \hookrightarrow^{m} \mathsf{v}$ *with* $n \geq m$, *then also* $\mathbf{t}\Downarrow$.

Proof. By Lemma 14 and Lemma 15. $\square$

Lemma 17 (Value relation implies oftype)**.** $\mathcal{V}[\![\hat{\tau}]\!]_{\square} \subseteq$ `oftype`$(\hat{\tau})$

Proof. Simple induction on $\hat{\tau}$. $\square$

5 Compiler

This section defines type erasure and protection for terms (Section 5.1), the two functions that constitute the compiler. Then it presents properties for erasure (Section 5.2) and for protection (Section 5.3). Finally it concludes with contextual equivalence reflection (Section 5.4).

Recall that we will use $\mathtt{b}$ to refer to $\mathtt{unit}$ / $\mathtt{unit}$, $\mathtt{true}$ / $\mathtt{true}$ and $\mathtt{false}$ / $\mathtt{false}$ when it is not necessary to specify or when it is obvious. Analogously, we use $\mathcal{B}$ to mean $\mathtt{Unit}$ or $\mathtt{Bool}$.

The compiler $[\![\cdot]\!]^{\lambda^\tau}_{\lambda^{\mathsf{u}}}$ is a function of type $\mathit{Terms}^{\lambda^\tau} \to \mathit{Terms}^{\lambda^{\mathsf{u}}}$ defined as follows:

$$\text{if } \mathbf{\Gamma} \vdash \mathbf{t} : \tau, \text{ then } [\![\mathbf{t}]\!]^{\lambda^\tau}_{\lambda^{\mathsf{u}}} \overset{\mathsf{def}}{=} \mathsf{protect}_\tau\ \mathtt{erase}(\mathbf{t})$$

Where $\mathtt{erase}(\cdot)$ is a function of type $\mathit{Terms}^{\lambda^\tau} \to \mathit{Terms}^{\lambda^{\mathsf{u}}}$ and $\mathsf{protect}_\tau$ is a λ^{u} term for any type τ.

5.1 Compiler definition: erase and protect

Function $\mathtt{erase}(\cdot)$ takes a λ^τ term and strips it of type annotations, effectively turning it into a λ^{u} term.

$$\mathtt{erase}(\mathbf{b}) = \mathsf{b} \qquad \mathtt{erase}(\lambda \mathbf{x} : \tau.\ \mathbf{t}) = \lambda \mathsf{x}.\ \mathtt{erase}(\mathbf{t})$$

$$\mathtt{erase}(\mathbf{x}) = \mathsf{x} \qquad \mathtt{erase}(\langle \mathbf{t_1}, \mathbf{t_2} \rangle) = \langle \mathtt{erase}(\mathbf{t_1}), \mathtt{erase}(\mathbf{t_2}) \rangle$$

$$\mathtt{erase}(\mathbf{t_1}\ \mathbf{t_2}) = \mathtt{erase}(\mathbf{t_1})\ \mathtt{erase}(\mathbf{t_2})$$

$$\mathtt{erase}(\mathbf{t}.\mathbf{1}) = \mathtt{erase}(\mathbf{t}).1 \qquad \mathtt{erase}(\mathbf{t}.\mathbf{2}) = \mathtt{erase}(\mathbf{t}).2$$

$$\mathtt{erase}(\mathrm{inl}\ \mathbf{t}) = \mathrm{inl}\ \mathtt{erase}(\mathbf{t}) \qquad \mathtt{erase}(\mathrm{inr}\ \mathbf{t}) = \mathrm{inr}\ \mathtt{erase}(\mathbf{t})$$

$$\mathtt{erase}(\mathbf{t_1}; \mathbf{t_2}) = \mathtt{erase}(\mathbf{t_1}); \mathtt{erase}(\mathbf{t_2})$$

$$\begin{aligned}
\mathtt{erase}(\mathrm{case}\ \mathbf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2}) &= \\
&\mathrm{case}\ \mathtt{erase}(\mathbf{t})\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x_1} \mapsto \mathtt{erase}(\mathbf{t_1}) \mid \mathrm{inr}\ \mathsf{x_2} \mapsto \mathtt{erase}(\mathbf{t_2}) \\
\mathtt{erase}(\mathrm{if}\ \mathbf{t}\ \mathrm{then}\ \mathbf{t_1}\ \mathrm{else}\ \mathbf{t_2}) &= \\
&\mathrm{if}\ \mathtt{erase}(\mathbf{t})\ \mathrm{then}\ \mathtt{erase}(\mathbf{t_1})\ \mathrm{else}\ \mathtt{erase}(\mathbf{t_2}) \\
\mathtt{erase}(\mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{t}) &= \mathit{fix}\ \mathtt{erase}(\mathbf{t})
\end{aligned}$$

For $\mathrm{fix}_{\tau_1 \to \tau_2}$ we use a strict fix combinator fix (Plotkin's Z combinator, see TAPL §5.2). We define

$$\mathit{fix} \overset{\mathsf{def}}{=} \lambda \mathsf{f}.\ (\lambda \mathsf{x}.\ \mathsf{f}\ (\lambda \mathsf{y}.\ \mathsf{x}\ \mathsf{x}\ \mathsf{y}))\ (\lambda \mathsf{x}.\ \mathsf{f}\ (\lambda \mathsf{y}.\ \mathsf{x}\ \mathsf{x}\ \mathsf{y}))$$

$$\mathit{fix}_{\mathsf{f}} \overset{\mathsf{def}}{=} (\lambda \mathsf{x}.\ \mathsf{f}\ (\lambda \mathsf{y}.\ \mathsf{x}\ \mathsf{x}\ \mathsf{y}))\ (\lambda \mathsf{x}.\ \mathsf{f}\ (\lambda \mathsf{y}.\ \mathsf{x}\ \mathsf{x}\ \mathsf{y}))$$

and we already note that if v is a value then

$$\mathit{fix}\ \mathsf{v} \hookrightarrow \mathit{fix}_{\mathsf{v}}$$

and we also have that

$$\mathit{fix}_{(\lambda \mathsf{x}.\, \mathsf{e})} \hookrightarrow (\lambda \mathsf{x}.\, \mathsf{e})\ (\lambda \mathsf{y}.\, \mathit{fix}_{\lambda \mathsf{x}.\mathsf{e}}\ \mathsf{y}) \hookrightarrow \mathsf{e}[(\lambda \mathsf{y}.\, \mathit{fix}_{\lambda \mathsf{x}.\mathsf{e}}\ \mathsf{y})/\mathsf{x}]$$

Function protect takes a λ^{τ} type to a function that wraps a term so that it behaves according to the type. The definition of protect relies on another function confine that is used to wrap externally-supplied parameters with the right checks that ensure no violation of source-level abstractions. Both functions are defined inductively on the type as presented below.

$$\begin{aligned}
\mathsf{protect}_{\mathcal{B}} &\overset{\mathsf{def}}{=} \lambda \mathsf{x}.\, \mathsf{x}\\
\mathsf{protect}_{\tau_1 \times \tau_2} &\overset{\mathsf{def}}{=} \lambda \mathsf{y}.\, \langle \mathsf{protect}_{\tau_1}\ \mathsf{y}.1, \mathsf{protect}_{\tau_2}\ \mathsf{y}.2 \rangle\\
\mathsf{protect}_{\tau_1 \uplus \tau_2} &\overset{\mathsf{def}}{=} \lambda \mathsf{y}.\, \text{case } \mathsf{y} \text{ of inl } \mathsf{x}_1 \mapsto \text{inl } (\mathsf{protect}_{\tau_1}\ \mathsf{x}_1) \mid \text{inr } \mathsf{x}_2 \mapsto \text{inr } (\mathsf{protect}_{\tau_2}\ \mathsf{x}_2)\\
\mathsf{protect}_{\tau_1 \to \tau_2} &\overset{\mathsf{def}}{=} \lambda \mathsf{y}.\, \lambda \mathsf{x}. \mathsf{protect}_{\tau_2}\ (\mathsf{y}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}))
\end{aligned}$$

$$\begin{aligned}
\mathsf{confine}_{\mathtt{Unit}} &\overset{\mathsf{def}}{=} \lambda \mathsf{y}.\, \mathsf{y}; \mathtt{unit}\\
\mathsf{confine}_{\mathtt{Bool}} &\overset{\mathsf{def}}{=} \lambda \mathsf{y}.\, \text{if } \mathsf{y} \text{ then } \mathtt{true} \text{ else } \mathtt{false}\\
\mathsf{confine}_{\tau_1 \times \tau_2} &\overset{\mathsf{def}}{=} \lambda \mathsf{y}.\, \langle \mathsf{confine}_{\tau_1}\ \mathsf{y}.1, \mathsf{confine}_{\tau_2}\ \mathsf{y}.2 \rangle\\
\mathsf{confine}_{\tau_1 \uplus \tau_2} &\overset{\mathsf{def}}{=} \lambda \mathsf{y}.\, \text{case } \mathsf{y} \text{ of inl } \mathsf{x}_1 \mapsto \text{inl } (\mathsf{confine}_{\tau_1}\ \mathsf{x}_1) \mid \text{inr } \mathsf{x}_2 \mapsto \text{inr } (\mathsf{confine}_{\tau_2}\ \mathsf{x}_2)\\
\mathsf{confine}_{\tau_1 \to \tau_2} &\overset{\mathsf{def}}{=} \lambda \mathsf{y}.\, \lambda \mathsf{x}.\, \mathsf{confine}_{\tau_2}\ (\mathsf{y}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x}))
\end{aligned}$$

The compiler security checks appear in the function type $\tau' \to \tau$ case for protect. There, we know that the term t will take an input and continue as a function. Therefore, the compiler wraps t in a function that takes the input, checks that it complies to τ', and then it supplies that input to $\mathbf{t}$. To check that an input complies to a type, confine is used. Dually, the function case for confine must call protect on the argument that in this case is supposedly coming from the compiled term.

The checks inserted for base types appear in the base type case `Bool` and `Unit` for confine. The returned argument, applied to the arguments supplied in the case of $\mathsf{confine}_{\mathcal{B}}$ ensures that if the argument t is not of base type, then the compiled term will diverge at runtime. If the argument t is of base type, then the execution will proceed normally.

5.2 Properties of erasure

This section presents required results (Lemmas 18 to 20). Then it presents compatibility lemmas (Lemmas 21 to 31 in Section 5.2.1). Finally, it concludes by proving semantics preservation of erase Theorems 4 and 5.

Lemma 18 (Erased contexts bind the same variables)**.** *If* $\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau$, *then* $\vdash \mathtt{erase}(\mathfrak{C}) : \mathtt{dom}(\mathbf{\Gamma}') \to \mathtt{dom}(\mathbf{\Gamma})$.

Proof. Trivial induction on $\mathbf{\Gamma}$. $\square$

Lemma 19 (Related terms plugged in related contexts are still related)**.** *If* $(\underline{\mathsf{W}}, \mathbf{t}, \mathsf{t}) \in \mathcal{E}[\![\hat{\tau}']\!]_{\square}$ *and if for all* $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\hat{\tau}']\!]_{\square}$, *we have that* $(\underline{\mathsf{W}}', \mathbb{E}[\mathbf{v}], \mathbb{E}[\mathsf{v}]) \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}$ *then* $(\underline{\mathsf{W}}, \mathbb{E}[\mathbf{t}], \mathbb{E}[\mathsf{t}]) \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}$.

Proof. Take $(\underline{\mathsf{W}}, \mathbb{E}', \mathbb{E}') \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$. It suffices to show that $(\mathbb{E}'[\mathbb{E}[\mathbf{t}]], \mathbb{E}'[\mathbb{E}[\mathsf{t}]]) \in \mathsf{O}(\underline{\mathsf{W}})$. This follows from $(\underline{\mathsf{W}}, \mathbf{t}, \mathsf{t}) \in \mathcal{E}[\![\hat{\tau}']\!]_{\square}$ if $(\underline{\mathsf{W}}, \mathbb{E}'[\mathbb{E}[\cdot]], \mathbb{E}'[\mathbb{E}[\cdot]]) \in \mathcal{K}[\![\hat{\tau}']\!]_{\square}$. So, take $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\hat{\tau}']\!]_{\square}$. We need to show that $(\mathbb{E}'[\mathbb{E}[\mathbf{v}]], \mathbb{E}'[\mathbb{E}[\mathsf{v}]]) \in \mathsf{O}(\underline{\mathsf{W}}')$. But this follows from $(\underline{\mathsf{W}}', \mathbb{E}[\mathbf{v}], \mathbb{E}[\mathsf{v}]) \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}$, since from $(\underline{\mathsf{W}}, \mathbb{E}', \mathbb{E}') \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$, we get $(\underline{\mathsf{W}}', \mathbb{E}', \mathbb{E}') \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$ by Lemma 12. $\square$

Lemma 20 (Related functions applied to related arguments are related terms)**.** *If* $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\hat{\tau}' \to \hat{\tau}]\!]_{\square}$ *and* $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\hat{\tau}']\!]_{\square}$ *then* $(\underline{\mathsf{W}}, \mathbf{v}\ \mathbf{v}', \mathsf{v}\ \mathsf{v}') \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}$.

Proof. Take $(\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$, then we need to show that $(\mathbb{E}[\mathbf{v}\ \mathbf{v}'], \mathbb{E}[\mathsf{v}\ \mathsf{v}']) \in \mathsf{O}(\underline{\mathsf{W}})$.

From $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\hat{\tau}' \to \hat{\tau}]\!]_{\square}$, we get that $\mathbf{v} \equiv \lambda\mathbf{x} : \hat{\tau}'.\ \mathbf{t}'$ and $\mathsf{v} \equiv \lambda\mathsf{x}.\ \mathsf{t}'$ for some $\mathbf{t}'$ and t'. We then know that $\mathbb{E}[\mathbf{v}\ \mathbf{v}'] \hookrightarrow \mathbb{E}[\mathbf{t}'[\mathbf{v}'/\mathbf{x}]]$ and $\mathbb{E}[\mathsf{v}_1\ \mathsf{v}_2] \hookrightarrow \mathbb{E}[\mathsf{t}'[\mathsf{v}_2/\mathsf{x}]]$ and by Lemma 8, it suffices to show that $(\mathbb{E}[\mathbf{t}'[\mathbf{v_2}/\mathbf{x}]], \mathbb{E}[\mathsf{t}'[\mathsf{v}'/\mathsf{x}]]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}})$.

Since $(\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$, $\triangleright \underline{\mathsf{W}} \sqsupseteq \underline{\mathsf{W}}$, we have by Lemma 12 that $(\triangleright \underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\hat{\tau}]\!]_{\square}$. It then suffices to prove that $(\triangleright \underline{\mathsf{W}}, \mathbf{t}'[\mathbf{v}'/\mathbf{x}], \mathsf{t}'[\mathsf{v}'/\mathsf{x}]) \in \mathcal{E}[\![\hat{\tau}]\!]_{\square}$. This follows from $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\hat{\tau}' \to \hat{\tau}]\!]_{\square}$, since $\triangleright \underline{\mathsf{W}} \sqsupset \underline{\mathsf{W}}$, if we show that $(\triangleright \underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\hat{\tau}']\!]_{\square}$. The latter follows from $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\hat{\tau}']\!]_{\square}$ by Lemma 13 since $\triangleright \underline{\mathsf{W}} \sqsupseteq \underline{\mathsf{W}}$. $\square$

5.2.1 Compatibility lemmas

Lemma 21 (Compatibility lemma for lambda)**.** *If* $\mathbf{\Gamma}, \mathbf{x} : \tau' \vdash \mathbf{t} \sqsupseteq_n \mathsf{t} : \tau$, *then* $\mathbf{\Gamma} \vdash \lambda\mathbf{x} : \tau'.\ \mathbf{t} \sqsupseteq_n \lambda\mathsf{x}.\mathsf{t} : \tau' \to \tau$.

Proof. By definition of $\sqsupseteq_n$, the thesis consists of two parts, which both must hold: (1) $\mathbf{\Gamma} \vdash \lambda\mathbf{x} : \tau'.\mathbf{t} : \tau' \to \tau$ and (2) for all $\underline{\mathsf{W}}$, $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$ (HG), we have that $(\underline{\mathsf{W}}, \lambda\mathbf{x} : \tau'.\mathbf{t}\gamma, \lambda\mathsf{x}.\mathsf{t}\gamma) \in \mathcal{E}[\![\tau' \to \tau]\!]_{\square}$.

Part 1 holds by the typing rule rule λ^{τ}-Type-fun combined with the fact $\mathbf{\Gamma}, \mathbf{x} : \tau' \vdash \mathbf{t} : \tau$ which we get from $\mathbf{\Gamma}, \mathbf{x} : \tau' \vdash \mathbf{t} \sqsupseteq_n \mathsf{t} : \tau$.

Let us now prove part 2.

By Lemma 10, it suffices to prove that $(\underline{\mathsf{W}}, \lambda\mathbf{x} : \tau'.\mathbf{t}\gamma, \lambda\mathsf{x}.\mathsf{t}\gamma) \in \mathcal{V}[\![\tau' \to \tau]\!]$.

Take $\underline{\mathsf{W}}' \sqsupset \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\tau']\!]$ (HV), then we need to show that $(\underline{\mathsf{W}}', \mathbf{t}\gamma[\mathbf{v}'/\mathbf{x}], \mathsf{t}\gamma[\mathsf{v}'/\mathsf{x}]) \in \mathcal{E}[\![\tau]\!]$.

The thesis follows from $\mathbf{\Gamma}, \mathbf{x} : \tau' \vdash \mathbf{t} \sqsupseteq_n \mathsf{t} : \tau$ if we show that $(\underline{\mathsf{W}}', [\mathbf{v}'/\mathbf{x}]\gamma, [\mathsf{v}'/\mathsf{x}]\gamma) \in \mathcal{G}[\![\mathbf{\Gamma}, (\mathbf{x} : \tau')]\!]$.

Unfold the definition of $\mathcal{G}[\![\mathbf{\Gamma}, (\mathbf{x} : \tau')]\!]_{\square}$, so the thesis becomes: (1) $(\underline{\mathsf{W}}', \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$ and (2) $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\tau']\!]_{\square}$.

Part 1 holds due to HG and Lemma 11, as HG holds in $\underline{\mathsf{W}}$ and here we need it in a future world $\underline{\mathsf{W}}'$.

Part 2 holds due to HV. $\square$

Lemma 22 (Compatibility lemma for pair)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \boldsymbol{\tau_1}$ *and IH2:* $\mathbf{\Gamma} \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \boldsymbol{\tau_2}$*, then* $\mathbf{\Gamma} \vdash \langle \mathbf{t_1}, \mathbf{t_2} \rangle \ \square_n \ \langle \mathsf{t}_1, \mathsf{t}_2 \rangle : \boldsymbol{\tau_1} \times \boldsymbol{\tau_2}$*.*

Proof. By definition of $\square_n$, the thesis consists of two parts, which both must hold: (1) $\mathbf{\Gamma} \vdash \langle \mathbf{t_1}, \mathbf{t_2} \rangle : \boldsymbol{\tau_1} \times \boldsymbol{\tau_2}$ and (2) for all $\underline{\mathsf{W}}$, $(\underline{\mathsf{W}}, \boldsymbol{\gamma}, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$, we have that $(\underline{\mathsf{W}}, \langle \mathbf{t_1}, \mathbf{t_2} \rangle \boldsymbol{\gamma}, \langle \mathsf{t}_1, \mathsf{t}_2 \rangle \gamma) \in \mathcal{E}[\![\boldsymbol{\tau_1} \times \boldsymbol{\tau_2}]\!]_{\square}$.

Part (1) holds by typing rule rule λ^{τ}-Type-pair and the facts that $\mathbf{\Gamma} \vdash \mathbf{t_1} : \boldsymbol{\tau_1}$ and $\mathbf{\Gamma} \vdash \mathbf{t_2} : \boldsymbol{\tau_2}$, which follow from $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \boldsymbol{\tau_1}$ and $\mathbf{\Gamma} \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \boldsymbol{\tau_2}$ respectively.

Let us now prove part (2). We have that $(\underline{\mathsf{W}}, \mathbf{t_1} \boldsymbol{\gamma}, \mathsf{t}_1 \gamma) \in \mathcal{E}[\![\boldsymbol{\tau_1}]\!]_{\square}$ from $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \boldsymbol{\tau_1}$. By Lemma 19, it then suffices to show that for all $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\boldsymbol{\tau_1}]\!]_{\square}$, we have that $(\underline{\mathsf{W}}', \langle \mathbf{v_1}, \mathbf{t_1} \boldsymbol{\gamma} \rangle, \langle \mathsf{v}_1, \mathsf{t}_2 \gamma \rangle) \in \mathcal{E}[\![\boldsymbol{\tau_1} \times \boldsymbol{\tau_2}]\!]_{\square}$.

From $\mathbf{\Gamma} \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \boldsymbol{\tau_2}$, we also have that $(\underline{\mathsf{W}}', \mathbf{t_2} \boldsymbol{\gamma}, \mathsf{t}_2 \gamma) \in \mathcal{E}[\![\boldsymbol{\tau_2}]\!]_{\square}$. Again by Lemma 19, it then suffices to show that for all $\underline{\mathsf{W}}'' \sqsupseteq \underline{\mathsf{W}}'$, $(\underline{\mathsf{W}}'', \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\boldsymbol{\tau_2}]\!]_{\square}$, we have that $(\underline{\mathsf{W}}'', \langle \mathbf{v_1}, \mathbf{v_2} \rangle, \langle \mathsf{v}_1, \mathsf{v}_2 \rangle) \in \mathcal{E}[\![\boldsymbol{\tau_1} \times \boldsymbol{\tau_2}]\!]_{\square}$.

By Lemma 10, it suffices to show that $(\underline{\mathsf{W}}'', \langle \mathbf{v_1}, \mathbf{v_2} \rangle, \langle \mathsf{v}_1, \mathsf{v}_2 \rangle) \in \mathcal{V}[\![\boldsymbol{\tau_1} \times \boldsymbol{\tau_2}]\!]_{\square}$, and the result follows by definition with $(\underline{\mathsf{W}}'', \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\boldsymbol{\tau_2}]\!]_{\square}$, $(\underline{\mathsf{W}}', \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\boldsymbol{\tau_1}]\!]_{\square}$ and using Lemma 13. $\square$

Lemma 23 (Compatibility lemma for application)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \tau' \to \tau$ *and IH2:* $\mathbf{\Gamma} \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \tau'$*, then* $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \mathbf{t_2} \ \square_n \ \mathsf{t}_1 \ \mathsf{t}_2 : \tau$*.*

Proof. By definition of $\square_n$, the thesis consists of two parts, which both must hold: (1) $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \mathbf{t_2} : \tau$ and (2) for all $\underline{\mathsf{W}}$, $(\underline{\mathsf{W}}, \boldsymbol{\gamma}, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$, we have that $(\underline{\mathsf{W}}, \mathbf{t_1} \boldsymbol{\gamma} \ \mathbf{t_2} \boldsymbol{\gamma}, \mathsf{t}_1 \gamma \ \mathsf{t}_2 \gamma) \in \mathcal{E}[\![\tau]\!]_{\square}$.

Part (1) holds because of the typing rule rule λ^{τ}-Type-app and the facts that $\mathbf{\Gamma} \vdash \mathbf{t_1} : \tau' \to \tau$ and $\mathbf{\Gamma} \vdash \mathbf{t_2} : \tau'$ which follow from $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \tau' \to \tau$ and $\mathbf{\Gamma} \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \tau'$ respectively.

Let us now prove part (2). We have that $(\underline{\mathsf{W}}, \mathbf{t_1} \boldsymbol{\gamma}, \mathsf{t}_1 \gamma) \in \mathcal{E}[\![\tau' \to \tau]\!]_{\square}$ from $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \tau' \to \tau$. By Lemma 19, it suffices to show that for all $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\tau' \to \tau]\!]_{\square}$, that $(\underline{\mathsf{W}}', \mathbf{v_1} \ \mathbf{t_2} \boldsymbol{\gamma}, \mathsf{v}_1 \ \mathsf{t}_2 \gamma) \in \mathcal{E}[\![\tau]\!]_{\square}$.

We also have that $(\underline{\mathsf{W}}', \mathbf{t_2} \boldsymbol{\gamma}, \mathsf{t}_2 \gamma) \in \mathcal{E}[\![\tau']\!]_{\square}$ from $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \mathbf{t_2} \ \square_n \ \mathsf{t}_1 \ \mathsf{t}_2 : \tau$. Again by Lemma 19, it suffices to show that for all $\underline{\mathsf{W}}'' \sqsupseteq \underline{\mathsf{W}}'$, $(\underline{\mathsf{W}}'', \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\tau']\!]_{\square}$, that $(\underline{\mathsf{W}}'', \mathbf{v_1} \ \mathbf{v_2}, \mathsf{v}_1 \ \mathsf{v}_2) \in \mathcal{E}[\![\tau]\!]_{\square}$.

From $(\underline{\mathsf{W}}', \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\tau' \to \tau]\!]_{\square}$, we get $(\underline{\mathsf{W}}'', \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\tau' \to \tau]\!]_{\square}$ by Lemma 13 and the result then follows by Lemma 20. $\square$

Lemma 24 (Compatibility lemma for left projection)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \boldsymbol{\tau_1} \times \boldsymbol{\tau_2}$*, then* $\mathbf{\Gamma} \vdash \mathbf{t_1.1} \ \square_n \ \mathsf{t}_1.1 : \boldsymbol{\tau_1}$*.*

Proof. By definition of $\square_n$, the thesis consists of two parts, which both must hold: (1) $\mathbf{\Gamma} \vdash \mathbf{t_1.1} : \boldsymbol{\tau_1}$ and (2) for all $\underline{\mathsf{W}}$, $(\underline{\mathsf{W}}, \boldsymbol{\gamma}, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$, we have that $(\underline{\mathsf{W}}, \mathbf{t_1.1} \boldsymbol{\gamma}, \mathsf{t}_1.1 \gamma) \in \mathcal{E}[\![\boldsymbol{\tau_1}]\!]_{\square}$.

Part (1) holds because of rule λ^{τ}-Type-proj1, and the fact that $\mathbf{\Gamma} \vdash \mathbf{t_1} : \boldsymbol{\tau_1} \times \boldsymbol{\tau_2}$, which follows from $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \boldsymbol{\tau_1} \times \boldsymbol{\tau_2}$.

Let us now prove part (2). We have that $(\underline{W}, \mathbf{t_1}\gamma, t_1\gamma) \in \mathcal{E}[\![\tau_1 \times \tau_2]\!]_\square$ from $\mathbf{\Gamma} \vdash \mathbf{t_1} \square_n t_1 : \tau_1 \times \tau_2$. By Lemma 19, the result follows if we prove that for all $\underline{W}' \sqsupseteq \underline{W}$, $(\underline{W}', \mathbf{v}, v) \in \mathcal{V}[\![\tau_1 \times \tau_2]\!]_\square$, we have that $(\underline{W}', \mathbf{v.1}, v.1) \in \mathcal{E}[\![\tau_1]\!]_\square$.

So, take $(\underline{W}', \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau_1]\!]_\square$, then we need to prove that $(\mathbb{E}[\mathbf{v.1}], \mathbb{E}[v.1]) \in \mathsf{O}(\underline{W}')$.

From $(\underline{W}', \mathbf{v}, v) \in \mathcal{V}[\![\tau_1 \times \tau_2]\!]_\square$, we know that $\mathbf{v} = \langle \mathbf{v_1}, \mathbf{v_2} \rangle$ and that $v = \langle v_1, v_2 \rangle$ for some $\mathbf{v_1}, \mathbf{v_2}, v_1, v_2$ with $(\underline{W}'', \mathbf{v_1}, v_1) \in \mathcal{V}[\![\tau_1]\!]_\square$ (HV) and $(\underline{W}'', \mathbf{v_2}, v_2) \in \mathcal{V}[\![\tau_2]\!]_\square$ for any $\underline{W}'' \sqsupset \underline{W}'$.

We have that $\mathbb{E}[\mathbf{v.1}] \hookrightarrow \mathbb{E}[\mathbf{v_1}]$ and $\mathbb{E}[v.1] \hookrightarrow \mathbb{E}[v_1]$, so by Lemma 8, it suffices to prove that $(\mathbb{E}[\mathbf{v_1}], \mathbb{E}[v_1]) \in \mathsf{O}(\triangleright \underline{W}')$. This follows because we know that $(\triangleright \underline{W}', \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau_1]\!]_\square$ from $(\underline{W}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau_1]\!]_\square$ and $\triangleright \underline{W}' \sqsupseteq \underline{W}$ by Lemma 12 and because we have that $(\triangleright \underline{W}', \mathbf{v_1}, v_1) \in \mathcal{V}[\![\tau_1]\!]_\square$ (HV). ☐

Lemma 25 (Compatibility lemma for right projection)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t_1} \square_n t_1 : \tau_1 \times \tau_2$*, then* $\mathbf{\Gamma} \vdash \mathbf{t_1.2} \square_n t_1.2 : \tau_2$*.*

Proof. Simple adaptation of the proof of Lemma 24. ☐

Lemma 26 (Compatibility lemma for inl)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t} \square_n t : \tau$ *then* $\mathbf{\Gamma} \vdash \text{inl } \mathbf{t} \square_n \text{inl } t : \tau \uplus \tau'$*.*

Proof. By definition of $\square_n$, the thesis consists of two parts, which both must hold: (1) $\mathbf{\Gamma} \vdash \text{inl } \mathbf{t} : \tau \uplus \tau'$ and (2) for all $\underline{W}$,$(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$, we have that $(\underline{W}, \text{inl } \mathbf{t}\gamma, \text{inl } t\gamma) \in \mathcal{E}[\![\tau \uplus \tau']\!]_\square$.

Part (1) holds by rule λ^τ-Type-inl and the fact that $\mathbf{\Gamma} \vdash \mathbf{t} : \tau$ which follows from $\mathbf{\Gamma} \vdash \mathbf{t} \square_n t : \tau$.

Let us now prove part (2). Expand the definition of $\square_n$. The thesis becomes $\forall (\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square, (\underline{W}, \text{inl } \mathbf{t}\gamma, \text{inl } t\gamma) \in \mathcal{E}[\![\tau \uplus \tau']\!]_\square$.

Expand the definition of $\mathcal{E}[\![\tau \uplus \tau']\!]_\square$. The thesis becomes $\forall (\underline{W}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau \uplus \tau']\!]_\square$ (HK), $(\mathbb{E}[\text{inl } \mathbf{t}\gamma], \mathbb{E}[\text{inl } t\gamma]) \in \mathsf{O}(\underline{W})$.

Take the hypothesis, expand the defintion of $\mathcal{E}[\![\tau]\!]_\square$ in it. We have that $\forall (\underline{W}', \gamma', \gamma') \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square, \forall (\underline{W}', \mathbb{E}', \mathbb{E}') \in \mathcal{K}[\![\tau]\!]_\square$, $(\mathbb{E}'[\mathbf{t}\gamma'], \mathbb{E}'[t\gamma']) \in \mathsf{O}(\underline{W}')$.

Instantiate $\underline{W}'$ with $\underline{W}$, $\mathbb{E}'$ with $\mathbb{E}[\text{inl } \cdot]$ and $\mathbb{E}'$ with $\mathbb{E}[\text{inl } \cdot]$.

The thesis is now proven, if we prove that $(\underline{W}, \mathbb{E}[\text{inl } \cdot], \mathbb{E}[\text{inl } \cdot]) \in \mathcal{K}[\![\tau]\!]_\square$.

Unfold the definition of $\mathcal{K}[\![\tau]\!]_\square$.

The thesis becomes $\forall \underline{W}' \sqsupseteq \underline{W}, \forall (\underline{W}', \mathbf{v}, v) \in \mathcal{V}[\![\tau]\!]_\square$ (HV), $(\mathbb{E}[\text{inl } \mathbf{v}], \mathbb{E}[\text{inl } v]) \in \mathsf{O}(\underline{W}')$.

Take HK and unfold the definition of $\mathcal{K}[\![\tau \uplus \tau']\!]_\square$.

We get that $\forall \underline{W}' \sqsupseteq \underline{W}, \forall (\underline{W}'', \mathbf{v}', v') \in \mathcal{V}[\![\tau \uplus \tau']\!]_\square$, $(\mathbb{E}[\mathbf{v}'], \mathbb{E}[v'] \in \mathsf{O}(\underline{W}'')$.

Instantiate $\underline{W}''$ with $\underline{W}'$ and $\mathbf{v}'$ with $\text{inl } \mathbf{v}$ and v' with $\text{inl } v$.

The thesis is now proven if we prove that $(\underline{W}', \text{inl } \mathbf{v}, \text{inl } v) \in \mathcal{V}[\![\tau \uplus \tau']\!]_\square$.

This follows from the definition of $\mathcal{V}[\![\tau \uplus \tau']\!]_\square$, given HV and Lemma 13 applied to HV. ☐

Lemma 27 (Compatibility lemma for inr)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t} \square_n t : \tau'$ *then* $\mathbf{\Gamma} \vdash \text{inr } \mathbf{t} \square_n \text{inr } t : \tau \uplus \tau'$*.*

Proof. Simple adaptation of the proof of Lemma 26. ☐

Lemma 28 (Compatibility lemma for case)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t} \ \square_n \ \mathsf{t} : \tau_1 \uplus \tau_2$ *(H),* $\mathbf{\Gamma}, (\mathbf{x_1} : \tau_1) \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \tau$ *(H1) and* $\mathbf{\Gamma}, (\mathbf{x_2} : \tau_2) \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \tau$ *(H2), then* $\mathbf{\Gamma} \vdash \text{case } \mathbf{t} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2} \ \square_n \ \text{case } \mathsf{t} \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2 : \tau$*.*

Proof. By definition of $\square_n$, the thesis consists of two parts, which both must hold: (1) $\mathbf{\Gamma} \vdash \text{case } \mathbf{t} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2} : \tau$ and (2) for all $\underline{\mathsf{W}}$,$(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$, we have that $(\underline{\mathsf{W}}, \text{case } \mathbf{t} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2}\gamma, \text{case } \mathsf{t} \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2\gamma) \in \mathcal{E}[\![\tau]\!]_{\square}$.

Part (1) holds by rule λ^{τ}-Type-case and the fact that $\mathbf{\Gamma} \vdash \mathbf{t} : \tau_1 \uplus \tau_2$ and $\mathbf{\Gamma}, (\mathbf{x_1} : \tau_1) \vdash \mathbf{t_1} : \tau$ and $\mathbf{\Gamma}, (\mathbf{x_2} : \tau_2) \vdash \mathbf{t_2} : \tau$ which follow from $\mathbf{\Gamma} \vdash \mathbf{t} \ \square_n \ \mathsf{t} : \tau_1 \uplus \tau_2$, $\mathbf{\Gamma}, (\mathbf{x_1} : \tau_1) \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \tau$ and $\mathbf{\Gamma}, (\mathbf{x_2} : \tau_2) \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \tau$.

Let us now prove part (2). Expand the definition of $\square_n$. The thesis becomes $\forall \underline{\mathsf{W}}, \forall (\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}, \forall (\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau]\!], (\mathbb{E}[\text{case } \mathbf{t} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2}], \mathbb{E}[\text{case } \mathsf{t} \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2]) \in \mathsf{O}(\underline{\mathsf{W}})$.

Expand H, we have that: $\forall \underline{\mathsf{W}}', \forall (\underline{\mathsf{W}}', \gamma', \gamma') \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$ (HG) $, \forall (\underline{\mathsf{W}}', \mathbb{E}', \mathbb{E}') \in \mathcal{K}[\![\tau_1 \uplus \tau_2]\!]_{\square}, (\mathbb{E}'[\mathbf{t}], \mathbb{E}'[\mathsf{t}]) \in \mathsf{O}(\underline{\mathsf{W}}')$.

Instantiate $\underline{\mathsf{W}}'$ with $\underline{\mathsf{W}}$, $\mathbb{E}' \cdot$ with $\mathbb{E}[\text{case } \cdot \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2}]$ and $\mathbb{E}' \cdot$ with $\mathbb{E}[\text{case } \cdot \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2]$.

The thesis holds if we prove that $(\underline{\mathsf{W}}, \mathbb{E}[\text{case } \cdot \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2}], \mathbb{E}[\text{case } \cdot \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2]) \in \mathcal{K}[\![\tau_1 \uplus \tau_2]\!]_{\square}$.

Unfold the definition of $\mathcal{K}[\![\tau_1 \uplus \tau_2]\!]_{\square}$.

The thesis becomes: $\forall \underline{\mathsf{W}}'' \sqsupseteq \underline{\mathsf{W}}, \forall (\underline{\mathsf{W}}'', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \uplus \tau_2]\!]_{\square}$ (HV) $, (\mathbb{E}[\text{case } \mathbf{v} \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2}], \mathbb{E}[\text{case } \mathsf{v} \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2]) \in \mathsf{O}(\underline{\mathsf{W}}'')$.

Unfold HV and the definition of $\mathcal{V}[\![\tau_1 \uplus \tau_2]\!]_{\square}$.

HV becomes $\mathsf{v} \in \mathsf{oftype}(\tau' \uplus \tau) \wedge \exists \mathbf{v}', \mathsf{v}'. \ ((\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \triangleright \mathcal{V}[\![\tau_1]\!]_{\square}$ (HV1) $\wedge (\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \square(\mathsf{WVRel}(\mathbf{inl}(\mathbf{v}'), \mathsf{inl}(\mathsf{v}'))))$ or $\exists \mathbf{v}', \mathsf{v}'. \ ((\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \triangleright \mathcal{V}[\![\tau_2]\!]_{\square} \wedge (\mathbf{v}, \mathsf{v}) \in \square(\mathsf{WVRel}(\underline{\mathsf{W}}, \mathbf{inr}(\mathbf{v}'), \mathsf{inr}(\mathsf{v}'))))$.

There are now 2 cases to consider: $\mathbf{v}$ and v being both inl or both inr .

inl Expand H1, we get: $\forall \underline{\mathsf{W}}_1, \forall (\underline{\mathsf{W}}_1, \gamma_1, \gamma_1) \in \mathcal{G}[\![\mathbf{\Gamma}, (\mathbf{x} : \tau_1)]\!], \forall (\underline{\mathsf{W}}_1, \mathbb{E}_1, \mathbb{E}_1) \in \mathcal{K}[\![\tau]\!]_{\square}, (\mathbb{E}_1[\mathbf{t_1}\gamma_1], \mathbb{E}_1[\mathsf{t}_1\gamma_1]) \in \mathsf{O}(\underline{\mathsf{W}}_1)$.

By definition of $\mathcal{G}[\![]\!]$, HG, Lemma 11, and HV1, we have that $(\triangleright \underline{\mathsf{W}}, [\mathbf{v}'/\mathbf{x_1}]\gamma, [\mathsf{v}'/\mathsf{x}_1]\gamma) \in \mathcal{G}[\![\mathbf{\Gamma}, (\mathbf{x} : \tau_1)]\!]$.

Therefore, we have that $(\mathbb{E}_1[\mathbf{t_1}[\mathbf{v}'/\mathbf{x_1}]\gamma], \mathbb{E}_1[\mathsf{t}_1[\mathsf{v}'/\mathsf{x}_1]\gamma]) \in \mathsf{O}(\underline{\mathsf{W}}_1)$.

We can apply Lemma 8 to prove the thesis.

In fact, rule λ^{τ}-Eval-case-inl tells us that $\mathbb{E}[\text{case inl } \mathbf{v}' \text{ of inl } \mathbf{x_1} \mapsto \mathbf{t_1} \mid \text{inr } \mathbf{x_2} \mapsto \mathbf{t_2}] \hookrightarrow \mathbb{E}[\mathbf{t_1}[\mathbf{v}/\mathbf{x_1}]]$, given that $\mathbf{v} \equiv \text{inl } \mathbf{v}'$.

And rule λ^{u}-Eval-case-inl tells us that $\mathbb{E}[\text{case inl } \mathsf{v}' \text{ of inl } \mathsf{x}_1 \mapsto \mathsf{t}_1 \mid \text{inr } \mathsf{x}_2 \mapsto \mathsf{t}_2] \hookrightarrow \mathbb{E}[\mathsf{t}_1[\mathsf{v}/\mathsf{x}_1]]$, given that $\mathsf{v} \equiv \text{inl } \mathsf{v}'$.

inr Analogous.

$\square$

Lemma 29 (Compatibility lemma for if)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \mathtt{Bool}$ *(H1) and* $\mathbf{\Gamma} \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \tau$ *(H2) and* $\mathbf{\Gamma} \vdash \mathbf{t_3} \ \square_n \ \mathsf{t}_3 : \tau$ *(H3), then* $\mathbf{\Gamma} \vdash \text{if } \mathbf{t_1} \text{ then } \mathbf{t_2} \text{ else } \mathbf{t_3} \ \square_n$ $\text{if } \mathsf{t}_1 \text{ then } \mathsf{t}_2 \text{ else } \mathsf{t}_3 : \tau$*.*

Proof. By definition of $\square_n$, the thesis consists of two parts, which both must hold: (1) $\mathbf{\Gamma} \vdash \text{if } \mathbf{t_1} \text{ then } \mathbf{t_2} \text{ else } \mathbf{t_3} : \tau$ and (2) for all $\underline{\mathsf{W}}$,$(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$, we have that $(\underline{\mathsf{W}}, \mathbf{t_1}\gamma; \mathbf{t_2}\gamma, \mathsf{t}_1\gamma; \mathsf{t}_2\gamma) \in \mathcal{E}[\![\tau]\!]_\square$.

Part (1) holds by rule λ^τ-Type-if and the fact that $\mathbf{\Gamma} \vdash \mathbf{t_1} : \mathtt{Bool}$ which follows from H1 and that $\mathbf{\Gamma} \vdash \mathbf{t_2} : \tau$ and $\mathbf{\Gamma} \vdash \mathbf{t_3} : \tau$ which follow from H2 and H3.

Let us now prove part (2). Expand the definition of $\square_n$ and of $\mathcal{E}[\![]\!]_\square$. The thesis becomes $\forall \underline{\mathsf{W}}$, $\forall (\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$, $\forall (\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau]\!]$, then $(\mathbb{E}[\text{if } \mathbf{t_1}\gamma \text{ then } \mathbf{t_2}\gamma \text{ else } \mathbf{t_3}\gamma]$, $\mathbb{E}[\text{if } \mathsf{t}_1\gamma \text{ then } \mathsf{t}_2\gamma \text{ else } \mathsf{t}_3\gamma]) \in \mathsf{O}(\underline{\mathsf{W}})$.

Unfold H1: $\forall \underline{\mathsf{W}}_1$, $\forall (\underline{\mathsf{W}}_1, \gamma_1, \gamma_1) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$, $\forall (\underline{\mathsf{W}}_1, \mathbb{E}_1, \mathbb{E}_1) \in \mathcal{K}[\![\mathtt{Bool}]\!]_\square$, $(\mathbb{E}_1[\mathbf{t_1}\gamma_1], \mathbb{E}_1[\mathsf{t}_1\gamma_1]) \in \mathsf{O}(\underline{\mathsf{W}}_1)$.

The thesis follows by instantiating $\underline{\mathsf{W}}_1$ with $\underline{\mathsf{W}}$, γ_1 with γ, γ_1 with γ and $\mathbb{E}_1$ with $\mathbb{E}[\text{if } [\cdot] \text{ then } \mathbf{t_2}\gamma \text{ else } \mathbf{t_3}\gamma]$ and $\mathbb{E}_1$ with $\mathbb{E}[\text{if } \cdot \text{ then } \mathsf{t}_2\gamma \text{ else } \mathsf{t}_3\gamma]$ if we prove that $(\underline{\mathsf{W}}, \mathbb{E}[\text{if } [\cdot] \text{ then } \mathbf{t_2}\gamma \text{ else } \mathbf{t_3}\gamma], \mathbb{E}[\text{if } \cdot \text{ then } \mathsf{t}_2\gamma \text{ else } \mathsf{t}_3\gamma]) \in \mathcal{K}[\![\mathtt{Bool}]\!]_\square$.

We expand the definition of $\mathcal{K}[\![]\!]_\square$ and the thesis becomes: $\forall \underline{\mathsf{W}}_f \sqsupseteq \underline{\mathsf{W}}, \forall (\underline{\mathsf{W}}_f, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{Bool}]\!]_\square$, $(\mathbb{E}[\text{if } \mathbf{v} \text{ then } \mathbf{t_2}\gamma \text{ else } \mathbf{t_3}\gamma], \mathbb{E}[\text{if } \mathsf{v} \text{ then } \mathsf{t}_2\gamma \text{ else } \mathsf{t}_3\gamma])\mathsf{O}(\underline{\mathsf{W}}_f)$.

We now have two cases: $\mathbf{v} \equiv \mathtt{true} \equiv \mathsf{v}$ or $\mathbf{v} \equiv \mathtt{false} \equiv \mathsf{v}$. We prove only the first, the second is analogous using H3 in place of H2.

Unfold H2. $\forall \underline{\mathsf{W}}_2$, $\forall (\underline{\mathsf{W}}_2, \gamma_2, \gamma_2) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$, $\forall (\underline{\mathsf{W}}_2, \mathbb{E}_2, \mathbb{E}_2) \in \mathcal{K}[\![\tau]\!]_\square$, $(\mathbb{E}_2[\mathbf{t_2}\gamma_2], \mathbb{E}_2[\mathsf{t}_2\gamma_2]) \in \mathsf{O}(\underline{\mathsf{W}}_2)$.

The thesis follows from Lemma 4 by rule λ^τ-Eval-if-v and rule λ^{u}-Eval-if-v since $\mathbf{v} \equiv \mathtt{true} \equiv \mathsf{v}$.

$\square$

Lemma 30 (Compatibility lemma for sequence)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \mathtt{Unit}$ *(H1) and* $\mathbf{\Gamma} \vdash \mathbf{t_2} \ \square_n \ \mathsf{t}_2 : \tau$ *(H2) then* $\mathbf{\Gamma} \vdash \mathbf{t_1}; \mathbf{t_2} \ \square_n \ \mathsf{t}_1; \mathsf{t}_2 : \tau$*.*

Proof. By definition of $\square_n$, the thesis consists of two parts, which both must hold: (1) $\mathbf{\Gamma} \vdash \mathbf{t_1}; \mathbf{t_2} : \tau$ and (2) for all $\underline{\mathsf{W}}$,$(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$, we have that $(\underline{\mathsf{W}}, \mathbf{t_1}\gamma; \mathbf{t_2}\gamma, \mathsf{t}_1\gamma; \mathsf{t}_2\gamma) \in \mathcal{E}[\![\tau]\!]$.

Part (1) holds by rule λ^τ-Type-seq and the fact that $\mathbf{\Gamma} \vdash \mathbf{t_1} : \mathtt{Unit}$ which follows from $\mathbf{\Gamma} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \mathtt{Unit}$.

Let us now prove part (2). Expand the definition of $\square_n$ and of $\mathcal{E}[\![]\!]_\square$. The thesis becomes $\forall \underline{\mathsf{W}}$, $\forall (\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$ (HK), $\forall (\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau]\!]_\square$, $(\mathbb{E}[\mathbf{t_1}\gamma; \mathbf{t_2}\gamma], \mathbb{E}[\mathsf{t}_1\gamma; \mathsf{t}_2\gamma]) \in \mathsf{O}(\underline{\mathsf{W}})$.

Unfold H1.

$\forall \underline{\mathsf{W}}_1$, $\forall (\underline{\mathsf{W}}_1, \gamma_1, \gamma_1) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$, $\forall (\underline{\mathsf{W}}_1, \mathbb{E}_1, \mathbb{E}_1) \in \mathcal{K}[\![\mathtt{Unit}]\!]_\square$, $(\mathbb{E}[\mathbf{t_1}\gamma_1], \mathbb{E}[\mathsf{t}_1\gamma_1]) \in \mathsf{O}(\underline{\mathsf{W}}_1)$.

The thesis holds by instiating $\underline{\mathsf{W}}_1$ with $\underline{\mathsf{W}}$, γ_1 with γ, γ_1 with γ, $\mathbb{E}_1$ with $\mathbb{E}[\cdot; \mathbf{t_2}\gamma]$ and $\mathbb{E}_1$ with $\mathbb{E}[\cdot; \mathsf{t}_2\gamma]$.

We need to prove that $(\underline{\mathsf{W}}, \mathbb{E}[\cdot; \mathbf{t_2}\gamma], \mathbb{E}[\cdot; \mathsf{t}_2\gamma]) \in \mathcal{K}[\![\mathtt{Unit}]\!]_\square$. The thesis is: $\forall \underline{\mathsf{W}}_f \sqsupseteq \underline{\mathsf{W}}$, $\forall (\underline{\mathsf{W}}_f, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{Unit}]\!]_\square$, $(\mathbb{E}[\mathbf{v}; \mathbf{t_2}\gamma], \mathbb{E}[\mathsf{v}; \mathsf{t}_2\gamma]) \in \mathsf{O}(\underline{\mathsf{W}}_f)$.

Assume $A = (\mathbb{E}[\mathbf{t_2}\gamma], \mathbb{E}[\mathsf{t}_2\gamma]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}}_f)$, the thesis follows from Lemma 4 because of rule λ^{τ}-Eval-seq-next and rule λ^{u}-Eval-seq-next and because $\mathbf{v} \equiv$ `unit` and $\mathsf{v} \equiv$ `unit`.

Prove A.

Unfold H2. $\forall \underline{\mathsf{W}}_2, \forall (\underline{\mathsf{W}}_2, \gamma_2, \gamma_2) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\Box, \forall (\underline{\mathsf{W}}_2, \mathbb{E}_2, \mathbb{E}_2) \in \mathcal{K}[\![\tau]\!]_\Box, (\mathbb{E}[\mathbf{t_2}\gamma_2], \mathbb{E}[\mathsf{t}_2\gamma_2]) \in \mathsf{O}(\underline{\mathsf{W}}_2)$.

The thesis follows by instantiating $\underline{\mathsf{W}}_2$ with $\triangleright \underline{\mathsf{W}}$, γ_2 with γ, γ_2 with γ and due to Lemma 12 applied to HK. $\Box$

Lemma 31 (Compatibility lemma for fix)**.** *If* $\mathbf{\Gamma} \vdash \mathbf{t} \,\Box_n\, \mathsf{t} : (\tau_1 \to \tau_2) \to \tau_1 \to \tau_2$*, then* $\mathbf{\Gamma} \vdash \mathrm{fix}_{\tau_1 \to \tau_2}\mathbf{t} \,\Box_n\, \mathit{fix}\ \mathsf{t} : \tau_1 \to \tau_2$*.*

For easy reference, we repeat the definition of fix*:*

$$\mathit{fix} \overset{\mathsf{def}}{=} \lambda \mathsf{f}.\,(\lambda \mathsf{x}.\,\mathsf{f}\ (\lambda \mathsf{y}.\,\mathsf{x}\ \mathsf{x}\ \mathsf{y}))\ (\lambda \mathsf{x}.\,\mathsf{f}\ (\lambda \mathsf{y}.\,\mathsf{x}\ \mathsf{x}\ \mathsf{y}))$$

Proof. Take $(\underline{\mathsf{W}}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau_1 \to \tau_2]\!]_\Box$. Then we need to prove that $(\mathbb{E}[\mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{t}\gamma], \mathbb{E}[\mathit{fix}\ \mathsf{t}\gamma]) \in \mathsf{O}(\underline{\mathsf{W}})_\Box$. Define $\mathbb{E}' \overset{\mathsf{def}}{=} \mathbb{E}[\mathrm{fix}_{\tau_1 \to \tau_2}\ \cdot]$ and $\mathbb{E}' \overset{\mathsf{def}}{=} \mathbb{E}[\mathit{fix}\ \cdot]$. The result follows from $\mathbf{\Gamma} \vdash \mathbf{t} \,\Box_n\, \mathsf{t} : (\tau_1 \to \tau_2) \to \tau_1 \to \tau_2$ if we prove that $(\underline{\mathsf{W}}, \mathbb{E}', \mathbb{E}') \in \mathcal{K}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\Box$.

So, take $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\Box$. Then we need to show that

$$(\mathbb{E}'[\mathbf{v}], \mathbb{E}'[\mathsf{v}]) = (\mathbb{E}[\mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}], \mathbb{E}[\mathit{fix}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}}')_\Box.$$

We have that $\mathbb{E}[\mathit{fix}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\mathit{fix}_{\mathsf{v}}]$, so by Lemma 4, it suffices to prove that

$$(\mathbb{E}[\mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}], \mathbb{E}[\mathit{fix}_{\mathsf{v}}]) \in \mathsf{O}(\underline{\mathsf{W}}')_\Box$$

or, sufficiently, $(\underline{\mathsf{W}}', \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}, \mathit{fix}_{\mathsf{v}}) \in \mathcal{E}[\![\tau_1 \to \tau_2]\!]_\Box$. We prove the latter for an arbitrary $\underline{\mathsf{W}}'$, by induction on $\mathsf{lev}(\underline{\mathsf{W}}')$, assuming that $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\Box$.

Take $(\underline{\mathsf{W}}', \mathbb{E}'', \mathbb{E}'') \in \mathcal{K}[\![\tau_1 \to \tau_2]\!]_\Box$, then we need to prove that $(\mathbb{E}''[\mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}], \mathbb{E}''[\mathit{fix}_{\mathsf{v}}]) \in \mathsf{O}(\underline{\mathsf{W}}')_\Box$. If $\mathsf{lev}(\underline{\mathsf{W}}') = 0$, then by Lemma 5, this is okay, so we assume that $\mathsf{lev}(\underline{\mathsf{W}}') > 0$. From $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\Box$, we get $\mathbf{t}$ and t such that $\mathbf{v} = \lambda \mathbf{x} : \tau_1 \to \tau_2.\, \mathbf{t}$ and $\mathsf{v} = \lambda \mathsf{x}.\, \mathsf{t}$. We have that $\mathbb{E}''[\mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}] \hookrightarrow \mathbb{E}''[\mathbf{t}[(\lambda\ \mathbf{y} : \tau_1.\ \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{y})/\mathbf{x}]]$ and $\mathbb{E}''[\mathit{fix}_{\mathsf{v}}] \hookrightarrow \mathbb{E}''[(\lambda \mathsf{x}.\ \mathsf{t})\ (\lambda \mathsf{y}.\, \mathit{fix}_{\mathsf{v}}\ \mathsf{y})] \hookrightarrow \mathbb{E}''[\mathsf{t}[(\lambda \mathsf{y}.\, \mathit{fix}_{\mathsf{v}}\ \mathsf{y})/\mathsf{x}]]$, and by Lemma 4, it suffices to prove that $(\mathbb{E}''[\mathbf{t}[(\lambda\ \mathbf{y} : \tau_1.\ \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{y})/\mathbf{x}]], \mathbb{E}''[\mathsf{t}[(\lambda \mathsf{y}.\, \mathit{fix}_{\mathsf{v}}\ \mathsf{y})/\mathsf{x}]]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}}')_\Box$. Note that since $\mathsf{lev}(\underline{\mathsf{W}}') > 0$, we have that $\mathsf{lev}(\triangleright \underline{\mathsf{W}}') < \mathsf{lev}(\underline{\mathsf{W}}')$.

First, we prove that

$$(\triangleright \underline{\mathsf{W}}', \lambda\ \mathbf{y} : \tau_1.\, \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{y}, \lambda \mathsf{y}.\, \mathit{fix}_{\mathsf{v}}\ \mathsf{y}) \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]_\Box.$$

By definition, this means proving, first, that $\emptyset \vdash \lambda\ \mathbf{y} : \tau_1.\, \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{y} : \tau_1 \to \tau_2$. We know from $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\Box$ that $\emptyset \vdash \mathbf{v} : (\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)$, from which this easily follows. Secondly, we need to prove that for all $\underline{\mathsf{W}}'' \sqsupseteq \triangleright \mathsf{W}'$, for all $(\underline{\mathsf{W}}'', \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\tau_1]\!]_\Box$, that $(\underline{\mathsf{W}}'', \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{v}', \mathit{fix}_{\mathsf{v}}\ \mathsf{v}') \in \mathcal{E}[\![\tau_2]\!]_\Box$. By induction on $\mathsf{lev}(\underline{\mathsf{W}}')$, we have that $(\underline{\mathsf{W}}'', \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}, \mathit{fix}_{\mathsf{v}}) \in \mathcal{E}[\![\tau_1 \to \tau_2]\!]_\Box$, since by monotonicity of $\mathcal{V}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\Box$, we know that $(\underline{\mathsf{W}}'', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\Box$. The result now follows directly by Lemmas 10 and 23.

Now that we have shown

$$(\triangleright \underline{\mathsf{W}}', \lambda\ \mathbf{y} : \tau_1.\ \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{y}, \lambda \mathsf{y}.\ \mathit{fix}_{\mathsf{v}}\ \mathsf{y}) \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]_\square,$$

we still need to show that $(\mathbb{E}''[\mathbf{t}[(\lambda\ \mathbf{y} : \tau_1.\ \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{y})/\mathbf{x}]], \mathbb{E}''[\mathsf{t}[(\lambda \mathsf{y}.\ \mathit{fix}_{\mathsf{v}}\ \mathsf{y})/\mathsf{x}]]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}}')_\square$. Since $\triangleright \underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}'$, we have that $(\triangleright \underline{\mathsf{W}}', \mathbb{E}'', \mathbb{E}'') \in \mathcal{K}[\![\tau_1 \to \tau_2]\!]_\square$ by Lemma 12. Therefore, it suffices to prove that $(\triangleright \underline{\mathsf{W}}', \mathbf{t}[(\lambda\ \mathbf{y} : \tau_1.\ \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{y})/\mathbf{x}], \mathsf{t}[(\lambda \mathsf{y}.\ \mathit{fix}_{\mathsf{v}}\ \mathsf{y})/\mathsf{x}]) \in \mathcal{E}[\![\tau_1 \to \tau_2]\!]_\square$. However, by definition of $\mathcal{V}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\square$, this follows directly from $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![(\tau_1 \to \tau_2) \to (\tau_1 \to \tau_2)]\!]_\square$, $\mathbf{v} = \lambda \mathbf{x} : \tau_1 \to \tau_2.\ \mathbf{t}$ and $\mathsf{v} = \lambda \mathsf{x}.\ \mathsf{t}$ and

$$(\triangleright \underline{\mathsf{W}}', \lambda\ \mathbf{y} : \tau_1.\ \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{v}\ \mathbf{y}, \lambda \mathsf{y}.\ \mathit{fix}_{\mathsf{v}}\ \mathsf{y}) \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]_\square.$$

□

Theorem 4 (Erase is semantics-preserving). *If* $\mathbf{\Gamma} \vdash \mathbf{t} : \tau$, *then* $\mathbf{\Gamma} \vdash \mathbf{t} \square_n \mathtt{erase}(\mathbf{t}) : \tau$ *for all* n.

Proof. The proof proceeds by induction on the type derivation of $\mathbf{\Gamma} \vdash \mathbf{t} : \tau$. The hypothesis H1 is that $\mathbf{\Gamma} \vdash \mathbf{t} : \tau$.

Rules λ^τ-unit to λ^τ-false Here, $\mathbf{t}$ is a primitive value $\mathtt{b}$ inhabiting type $\mathcal{B}$.

The thesis is: $\mathbf{\Gamma} \vdash \mathtt{b} \square_n \mathtt{erase}(\mathtt{b}) : \mathcal{B}$.

By applying $\mathtt{erase}(\cdot)$, the thesis becomes: $\mathbf{\Gamma} \vdash \mathtt{b} \square_n \mathtt{b} : \mathcal{B}$.

By definition of $\square_n$, the thesis consists of 2 parts, which both must hold: (1)$\mathbf{\Gamma} \vdash \mathtt{b} : \mathcal{B} \wedge$ (2)$\forall \underline{\mathsf{W}}, \forall (\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square, (\underline{\mathsf{W}}, \mathtt{b}\gamma, \mathtt{b}\gamma) \in \mathcal{E}[\![\mathcal{B}]\!]_\square$

Part 1 holds because of hypothesis H1.

For part 2, note that substitutions ($\gamma$and γ) do not affect $\mathtt{b}$.

Part 2 becomes: $\forall \underline{\mathsf{W}}, (\underline{\mathsf{W}}, \mathtt{b}, \mathtt{b}) \in \mathcal{E}[\![\mathcal{B}]\!]_\square$.

By Lemma 10, it suffices to prove that $(\underline{\mathsf{W}}, \mathtt{b}, \mathtt{b}) \in \mathcal{V}[\![\mathcal{B}]\!]_\square$, which is true by definition.

Rule λ^τ-Type-var Here, $\mathbf{t}$ is a variable $\mathbf{x}$.

The thesis is: $\mathbf{\Gamma} \vdash \mathbf{x} \square_n \mathtt{erase}(\mathbf{x}) : \tau$.

By applying $\mathtt{erase}(\cdot)$, the thesis becomes: $\mathbf{\Gamma} \vdash \mathbf{x} \square_n \mathsf{x} : \tau$.

By definition of $\square_n$, the thesis consists of 2 parts, which both must hold: (1)$\mathbf{\Gamma} \vdash \mathbf{x} : \tau \wedge$ (2)$\forall \underline{\mathsf{W}}, \forall (\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square, (\underline{\mathsf{W}}, \mathbf{x}\gamma, \mathsf{x}\gamma) \in \mathcal{E}[\![\tau]\!]_\square$.

Part 1 holds because of hypothesis H1.

Let us now prove part 2.

By H1 we know that $\mathbf{x} \in \mathtt{dom}(\mathbf{\Gamma})$.

By the definition of $\mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$, we know that $\mathbf{x} \in \mathtt{dom}(\gamma)$, that $\mathsf{x} \in \mathtt{dom}(\gamma)$, that we can replace $\mathbf{x}\gamma$ with $\mathbf{v}$ and $\mathsf{x}\gamma$ with v and that $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau]\!]_\square$ (HV).

This case holds by applying Lemma 10 to HV.

Rule λ^τ**-Type-fun** Here, $\mathbf{t}$ is a lambda-abstraction of the form $\lambda\mathbf{x} : \tau'.\mathbf{t}$ while τ is an arrow type of the form $\tau' \to \tau$.

The thesis is: $\mathbf{\Gamma} \vdash \lambda\mathbf{x} : \tau'.\mathbf{t} \;\square_n\; \mathtt{erase}(\lambda\mathbf{x} : \tau'.\mathbf{t}) : \tau' \to \tau$.

The inductive hypothesis IH is $\mathbf{\Gamma}, (\mathbf{x} : \tau') \vdash \mathbf{t} \;\square_n\; \mathtt{erase}(\mathbf{t}) : \tau$.

The result follows from Lemma 21, since $\mathtt{erase}(\lambda\mathbf{x} : \tau'.\mathbf{t}) = \lambda\mathsf{x}.\ \mathtt{erase}(\mathbf{t})$.

Rule λ^τ**-Type-pair** Here, $\mathbf{t}$ is a pair of the form $\langle\mathbf{t_1}, \mathbf{t_2}\rangle$ while τ is a product type of the form $\tau_1 \times \tau_2$.

The thesis is: $\mathbf{\Gamma} \vdash \langle\mathbf{t_1}, \mathbf{t_2}\rangle \;\square_n\; \mathtt{erase}(\langle\mathbf{t_1}, \mathbf{t_2}\rangle) : \tau_1 \times \tau_2$.

There are two inductive hypotheses IH1: $\mathbf{\Gamma} \vdash \mathbf{t_1} \;\square_n\; \mathtt{erase}(\mathbf{t_1}) : \tau_1$ and IH2: $\mathbf{\Gamma} \vdash \mathbf{t_2} \;\square_n\; \mathtt{erase}(\mathbf{t_2}) : \tau_2$.

The result follows from Lemma 22, since $\mathtt{erase}(\langle\mathbf{t_1}, \mathbf{t_2}\rangle) = \langle\mathtt{erase}(\mathbf{t_1}), \mathtt{erase}(\mathbf{t_2})\rangle$.

Rule λ^τ**-Type-app** Here, $\mathbf{t}$ is $\mathbf{t_1}\ \mathbf{t_2}$.

The thesis is $\mathbf{\Gamma} \vdash \mathbf{t_1}\ \mathbf{t_2} \;\square_n\; \mathtt{erase}(\mathbf{t_1}\ \mathbf{t_2}) : \tau$.

We have two inductive hypotheses: IH1 $= \mathbf{\Gamma} \vdash \mathbf{t_1} \;\square_n\; \mathtt{erase}(\mathbf{t_1}) : \tau' \to \tau$ and IH2 $= \mathbf{\Gamma} \vdash \mathbf{t_2} \;\square_n\; \mathtt{erase}(\mathbf{t_2}) : \tau'$.

The result follows from Lemma 23, since $\mathtt{erase}(\mathbf{t_1}\ \mathbf{t_2}) = \mathtt{erase}(\mathbf{t_1})\ \mathtt{erase}(\mathbf{t_2})$.

Rule λ^τ**-Type-proj1** Here, $\mathbf{t}$ is $\mathbf{t_1}.\mathbf{1}$ while τ is τ_1.

The thesis is $\mathbf{\Gamma} \vdash \mathbf{t_1}.\mathbf{1} \;\square_n\; \mathtt{erase}(\mathbf{t_1}.\mathbf{1}) : \tau_1$.

There is one inductive hypothesis IH: $\mathbf{\Gamma} \vdash \mathbf{t_1} \;\square_n\; \mathtt{erase}(\mathbf{t_1}) : \tau_1 \times \tau_2$.

The result follows from Lemma 24, since $\mathtt{erase}(\mathbf{t_1}.\mathbf{1}) = \mathtt{erase}(\mathbf{t_1}).1$.

Rule λ^τ**-Type-inl** Here, $\mathbf{t}$ is $\mathrm{inl}\ \mathbf{t_1}$ while τ is τ_1.

The thesis is $\mathbf{\Gamma} \vdash \mathrm{inl}\ \mathbf{t_1} \;\square_n\; \mathtt{erase}(\mathrm{inl}\ \mathbf{t_1}) : \tau_1 \uplus \tau_2$.

There is one inductive hypothesis IH: $\mathbf{\Gamma} \vdash \mathbf{t_1} \;\square_n\; \mathtt{erase}(\mathbf{t_1}) : \tau_1$.

The result follows from Lemma 26, since $\mathtt{erase}(\mathrm{inl}\ \mathbf{t_1}) = \mathrm{inl}\ \mathtt{erase}(\mathbf{t_1})$.

Rule λ^τ**-Type-inr** Here, $\mathbf{t}$ is $\mathrm{inr}\ \mathbf{t_2}$ while τ is τ_2.

The thesis is $\mathbf{\Gamma} \vdash \mathrm{inr}\ \mathbf{t_2} \;\square_n\; \mathtt{erase}(\mathrm{inr}\ \mathbf{t_2}) : \tau_1 \uplus \tau_2$.

There is one inductive hypothesis IH: $\mathbf{\Gamma} \vdash \mathbf{t_2} \;\square_n\; \mathtt{erase}(\mathbf{t_2}) : \tau_2$.

The result follows from Lemma 27, since $\mathtt{erase}(\mathrm{inr}\ \mathbf{t_2}) = \mathrm{inl}\ \mathtt{erase}(\mathbf{t_2})$.

Rule λ^τ**-Type-case** Here, $\mathbf{t}$ is $\mathrm{case}\ \mathbf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2}$ while τ is $\tau_1 \uplus \tau_2$.

The thesis is $\mathbf{\Gamma} \vdash \mathrm{case}\ \mathbf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2} \;\square_n\; \mathtt{erase}(\mathrm{case}\ \mathbf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2}) : \tau_1 \uplus \tau_2$.

There are three inductive hypotheses: $\mathbf{\Gamma} \vdash \mathbf{t} \;\square_n\; \mathtt{erase}(\mathbf{t}) : \tau_1 \uplus \tau_2$, $\mathbf{\Gamma}, (\mathbf{x_1} : \tau_1) \vdash \mathbf{t_1} \;\square_n\; \mathtt{erase}(\mathbf{t_1}) : \tau_1$ and $\mathbf{\Gamma}, (\mathbf{x_2} : \tau_2) \vdash \mathbf{t_2} \;\square_n\; \mathtt{erase}(\mathbf{t_2}) : \tau_2$.

The result follows from Lemma 28, since $\mathtt{erase}(\mathrm{case}\ \mathbf{t}\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x_1} \mapsto \mathbf{t_1} \mid \mathrm{inr}\ \mathbf{x_2} \mapsto \mathbf{t_2}) = \mathrm{case}\ \mathtt{erase}(\mathbf{t})\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x}_1 \mapsto \mathtt{erase}(\mathbf{t_1}) \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathtt{erase}(\mathbf{t_2})$.

Rule λ^τ-Type-fix We have that $\mathbf{t} = \mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{e}'$. $\mathtt{erase}(\mathrm{fix}_{\tau_1 \to \tau_2}\ \mathbf{e}') = \mathit{fix}\ \mathtt{erase}(\mathbf{e}')$.

The result follows from the induction hypothesis and Lemma 31.

Rule λ^τ-Type-if We have that $\mathbf{t} = \text{if } \mathbf{t}' \text{ then } \mathbf{t_1} \text{ else } \mathbf{t_2}$. $\mathtt{erase}(\text{if } \mathbf{t}' \text{ then } \mathbf{t_1} \text{ else } \mathbf{t_2}) = \text{if } \mathtt{erase}(\mathbf{t}') \text{ then } \mathtt{erase}(\mathbf{t_1}) \text{ else } \mathtt{erase}(\mathbf{t_2})$

The result follows from the induction hypotheses and Lemma 29.

Rule λ^τ-Type-seq : We have that $\mathbf{t} = \mathbf{t}; \mathbf{t}'$. $\mathtt{erase}(\mathbf{t}; \mathbf{t}') = \mathtt{erase}(\mathbf{t});\mathtt{erase}(\mathbf{t}')$

The result follows from the induction hypotheses and Lemma 30.

∎

Theorem 5 (Erasure is semantics preserving for contexts). *For all $\mathfrak{C}$, if $\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau$ then $\vdash \mathfrak{C}\ \square_n\ \mathtt{erase}(\mathfrak{C}) : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau$.*

Proof. Take $\mathbf{t}$,t with $\mathbf{\Gamma}' \vdash \mathbf{t}\ \square_n\ \mathsf{t} : \tau'$. Then we need to show that $\mathbf{\Gamma} \vdash \mathfrak{C}[\mathbf{t}]\ \square_n\ \mathtt{erase}(\mathfrak{C})[\mathsf{t}] : \tau$. We do this by induction on $\vdash \mathfrak{C} : \mathbf{\Gamma}', \tau' \to \mathbf{\Gamma}, \tau$.

The case for λ^τ-Type-Ctx-Hole is tautological. The other cases follow easily using the compatibility lemmas: Lemmas 21 to 31. ∎

5.3 Properties of dynamic type wrappers

This section proves additional results and then that protect is semantics preserving Theorem 6.

Lemma 32 (Protected and confineed terms reduce). *If $\mathsf{v} \in \mathsf{oftype}(\tau)$, then there exists a v' such that $\mathbb{E}[\mathsf{protect}_\tau\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}']$ for any $\mathbb{E}$ and $\mathsf{v}' \in \mathsf{oftype}(\tau)$ and there exists a v'' such that $\mathbb{E}[\mathsf{confine}_\tau\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}'']$ for any $\mathbb{E}$ and $\mathsf{v}'' \in \mathsf{oftype}(\tau)$.*

Proof. By induction on τ.

- $\tau = \mathcal{B}$ for some $\mathcal{B}$: For any $\mathbb{E}$, we have that $\mathbb{E}[\mathsf{protect}_\mathcal{B}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\mathsf{v}]$. We already know that $\mathsf{v} \in \mathsf{oftype}(\mathtt{Bool})$.

  For $\mathcal{B} = \mathtt{Unit}$, we have that

  $$\mathbb{E}[\mathsf{confine}_{\mathtt{Unit}}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\mathsf{v}; \mathtt{unit}]$$

  From $\mathsf{v} \in \mathsf{oftype}(\mathtt{Unit})$, we get that $\mathsf{v} = \mathtt{unit}$, from which we get that

  $$\mathbb{E}[\mathsf{v}; \mathtt{unit}] \hookrightarrow \mathbb{E}[\mathsf{v}]$$

  We already know that $\mathsf{v} \in \mathsf{oftype}(\mathtt{Unit})$.

  For $\mathcal{B} = \mathtt{Bool}$, we have that

  $$\mathbb{E}[\mathsf{confine}_{\mathtt{Bool}}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\text{if } \mathsf{v} \text{ then } \mathtt{true} \text{ else } \mathtt{false}]$$

  From $\mathsf{v} \in \mathsf{oftype}(\mathtt{Bool})$, we get that $\mathsf{v} = \mathtt{true}$ or $\mathsf{v} = \mathtt{false}$, from which we get that

  $$\mathbb{E}[\text{if } \mathsf{v} \text{ then } \mathtt{true} \text{ else } \mathtt{false}] \hookrightarrow \mathbb{E}[\mathsf{v}]$$

  We already know that $\mathsf{v} \in \mathsf{oftype}(\mathtt{Bool})$.

- $\tau = \tau_1 \times \tau_2$: By definition of $\mathsf{oftype}(\tau_1 \times \tau_2)$, we have that $\mathsf{v} = \langle \mathsf{v}_1, \mathsf{v}_2 \rangle$ with $\mathsf{v}_1 \in \mathsf{oftype}(\tau_1)$ and $\mathsf{v}_2 \in \mathsf{oftype}(\tau_2)$.

  For any $\mathbb{E}$, we have that

$$\begin{aligned}\mathbb{E}[\mathsf{protect}_{\tau_1 \times \tau_2}\ \mathsf{v}] &\hookrightarrow \mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v.1}, \mathsf{protect}_{\tau_2}\ \mathsf{v.2} \rangle] \hookrightarrow \\ &\mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}_1, \mathsf{protect}_{\tau_2}\ \mathsf{v.2} \rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}'_1, \mathsf{protect}_{\tau_2}\ \mathsf{v.2} \rangle] \hookrightarrow \\ &\qquad \mathbb{E}[\langle \mathsf{v}'_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}_2 \rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}'_1, \mathsf{v}'_2 \rangle]\end{aligned}$$

  where we use the induction hypotheses to obtain v'_1 and v'_2 such that the relevant parts of the above evaluation hold. The fact that $\langle \mathsf{v}'_1, \mathsf{v}'_2 \rangle \in \mathsf{oftype}(\tau_1 \times \tau_2)$ follows from the definition and the corresponding results of the induction hypotheses.

  The proof for $\mathsf{confine}_{\tau_1 \times \tau_2}$ is symmetric.

- $\tau = \tau_1 \uplus \tau_2$: By definition of $\mathsf{oftype}(\tau_1 \uplus \tau_2)$, we have that $\mathsf{v} = \mathrm{inl}\ \mathsf{v}_1$ with $\mathsf{v}_1 \in \mathsf{oftype}(\tau_1)$ or $\mathsf{v} = \mathrm{inr}\ \mathsf{v}_1$ with $\mathsf{v}_2 \in \mathsf{oftype}(\tau_2)$. We give the proof for the first case, the other case is similar.

  For any $\mathbb{E}$, we have that

$$\begin{aligned}&\mathbb{E}[\mathsf{protect}_{\tau_1 \uplus \tau_2}\ \mathsf{v}] \hookrightarrow \\ &\mathbb{E}[\mathrm{case}\ \mathsf{v}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x}_1 \mapsto \mathrm{inl}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x}_1) \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathrm{inr}\ (\mathsf{protect}_{\tau_2}\ \mathsf{x}_2)] \hookrightarrow \\ &\qquad \mathbb{E}[\mathrm{inl}\ (\mathsf{protect}_{\tau_1}\ \mathsf{v}_1)] \hookrightarrow \mathbb{E}[\mathrm{inl}\ \mathsf{v}'_1]\end{aligned}$$

  where we use the induction hypotheses to obtain a v'_1 such that the relevant part of the above evaluation holds. The fact that $\mathrm{inl}\ \mathsf{v}'_1 \in \mathsf{oftype}(\tau_1 \uplus \tau_2)$ follows from the definition and the corresponding result of the induction hypothesis.

  The proof for $\mathsf{confine}_{\tau_1 \uplus \tau_2}$ is symmetric.

- $\tau = \tau_1 \to \tau_2$: For any $\mathbb{E}$, we have that

$$\mathbb{E}[\mathsf{protect}_{\tau_1 \to \tau_2}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\lambda \mathsf{x}.\mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}))]$$

  and

$$\mathbb{E}[\mathsf{confine}_{\tau_1 \to \tau_2}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\lambda \mathsf{x}.\mathsf{confine}_{\tau_2}\ (\mathsf{v}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x}))].$$

  The fact that $\lambda \mathsf{x}.\mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x})) \in \mathsf{oftype}(\tau_1 \to \tau_2)$ and $\lambda \mathsf{x}.\mathsf{confine}_{\tau_2}\ (\mathsf{v}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x})) \in \mathsf{oftype}(\tau_1 \to \tau_2)$ follows from the definition.

$\square$

Lemma 33 (Related protected terms reduce and they are still related)**.** *For any τ,*

If $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau]\!]_{\square}$, then

- *there exists a* v' *such that* $\mathbb{E}[\mathsf{protect}_{\tau}\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}']$ *for any context* $\mathbb{E}$ *and* $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}') \in \mathcal{V}[\![\tau]\!]_{\square}$.
- *there exists a* v'' *such that* $\mathbb{E}[\mathsf{confine}_{\tau}\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}'']$ *for any context* $\mathbb{E}$ *and* $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}'') \in \mathcal{V}[\![\tau]\!]_{\square}$.

Proof. We prove this by induction on τ.

- $\tau = \mathcal{B}$: We have that $\mathsf{protect}_{\mathcal{B}} = \lambda \mathsf{y}.\, \mathsf{y}$ and

$$\begin{aligned}\mathsf{confine}_{\mathtt{Unit}} &= \lambda \mathsf{y}.\, \mathsf{y}; \mathtt{unit}\\ \mathsf{confine}_{\mathtt{Bool}} &= \lambda \mathsf{y}.\, \mathrm{if}\ \mathsf{y}\ \mathrm{then}\ \mathtt{true}\ \mathrm{else}\ \mathtt{false}\end{aligned}$$

  From $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{Unit}]\!]_{\square}$, we get that $\mathbf{v} = \mathsf{v} = \mathtt{unit}$ and from $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{Bool}]\!]_{\square}$, we get that $\mathbf{v} = \mathsf{v} = v$ with $v \in \{\mathtt{true}, \mathtt{false}\}$.

  For $\mathsf{protect}_{\mathcal{B}}$, it's clear that $\mathbb{E}[\mathsf{protect}_{\mathcal{B}}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\mathsf{v}]$ and that $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$.

  For $\mathsf{confine}_{\mathcal{B}}$, we can prove in all cases that

  $$\mathbb{E}[\mathsf{confine}_{\mathcal{B}}\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}]$$

  and it is clear that $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$.

- $\tau = \tau_1 \to \tau_2$: We have (by definition) that

  $$\mathsf{protect}_{\tau_1 \to \tau_2} = \lambda \mathsf{y}.\, \lambda \mathsf{x}. \mathsf{protect}_{\tau_2}\ (\mathsf{y}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}))$$

  and

  $$\mathsf{confine}_{\tau_1 \to \tau_2} = \lambda \mathsf{y}.\, \lambda \mathsf{x}.\, \mathsf{confine}_{\tau_2}\ (\mathsf{y}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x})).$$

  We do the proof for $\mathsf{protect}_{\tau_1 \to \tau_2}$, the proof for $\mathsf{confine}_{\tau_1 \to \tau_2}$ is symmetric.

  We have that $\mathbb{E}[\mathsf{protect}_{\tau_1 \to \tau_2}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\lambda \mathsf{x}.\ \mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}))]$. Now we need to prove that $(\underline{\mathsf{W}}, \mathbf{v}, \lambda \mathsf{x}.\ \mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}))) \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]_{\square}$.

  From $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]_{\square}$, we have that $\emptyset \vdash \mathbf{v} : \tau_1 \to \tau_2$, and that there exist $\mathbf{t}$ and t such that $\mathbf{v} = \lambda \mathbf{x} : \tau_1.\, \mathbf{t}$ and $\mathsf{v} = \lambda \mathsf{x}.\, \mathsf{t}$. It remains to prove that for any $\underline{\mathsf{W}}' \sqsupset \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\tau_1]\!]_{\square}$, we have that $(\underline{\mathsf{W}}', \mathbf{t}[\mathbf{v}'/\mathbf{x}], \mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{v}'))) \in \mathcal{E}[\![\tau_2]\!]_{\square}$.

  So, take $(\underline{\mathsf{W}}', \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau_2]\!]_{\square}$, then we need to prove

  $$(\mathbb{E}[\mathbf{t}[\mathbf{v}'/\mathbf{x}]], \mathbb{E}[\mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{v}'))]) \in \mathsf{O}(\underline{\mathsf{W}}')_{\square}.$$

  Since $\mathbb{E}[\mathsf{protect}_{\tau_2}\ (\mathsf{v}\ \cdot)]$ is an evaluation context and $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\tau_1]\!]_{\square}$, we have by induction that

  $$\mathbb{E}[\mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{v}'))] \hookrightarrow^* \mathbb{E}[\mathsf{protect}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}'')]$$

for some v'' such that $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}'') \in \mathcal{V}[\![\tau_1]\!]_\square$. By Lemma 4, it suffices to prove that

$$(\mathbb{E}[\mathbf{t}[\mathbf{v}'/\mathbf{x}]], \mathbb{E}[\mathsf{protect}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}'')]) \in \mathsf{O}(\underline{\mathsf{W}}')_\square.$$

Furthermore, we have that

$$\mathbb{E}[\mathsf{protect}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}'')] \hookrightarrow \mathbb{E}[\mathsf{protect}_{\tau_2}\ (\mathsf{t}[\mathsf{v}''/\mathsf{x}])]$$

and again by Lemma 4, it suffices to prove that

$$(\mathbb{E}[\mathbf{t}[\mathbf{v}'/\mathbf{x}]], \mathbb{E}[\mathsf{protect}_{\tau_2}\ (\mathsf{t}[\mathsf{v}''/\mathsf{x}])]) \in \mathsf{O}(\underline{\mathsf{W}}')_\square.$$

From $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]_\square$ and $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}'') \in \mathcal{V}[\![\tau_1]\!]_\square$, we have that $(\underline{\mathsf{W}}', \mathbf{t}[\mathbf{v}'/\mathbf{x}], \mathsf{t}[\mathsf{v}''/\mathsf{x}]) \in \mathcal{E}[\![\tau_2]\!]_\square$. It then suffices to prove that $(\underline{\mathsf{W}}', \mathbb{E}, \mathbb{E}[\mathsf{protect}_{\tau_2}\ \cdot]) \in \mathcal{K}[\![\tau_2]\!]_\square$.

So, take $\underline{\mathsf{W}}'' \sqsupseteq \underline{\mathsf{W}}'$ and $(\underline{\mathsf{W}}'', \mathbf{v}''', \mathsf{v}''') \in \mathcal{V}[\![\tau_2]\!]_\square$. Then it suffices to prove that $(\mathbb{E}[\mathbf{v}'''], \mathbb{E}[\mathsf{protect}_{\tau_2}\ \mathsf{v}''']) \in \mathsf{O}(\underline{\mathsf{W}}'')_\square$. Again, we have by induction that $\mathbb{E}[\mathsf{protect}_{\tau_2}\ \mathsf{v}'''] \hookrightarrow^* \mathbb{E}[\mathsf{v}'''']$ for some v'''' with $(\underline{\mathsf{W}}'', \mathbf{v}''', \mathsf{v}'''') \in \mathcal{V}[\![\tau_2]\!]_\square$. By Lemma 4, it suffices to prove that $(\mathbb{E}[\mathbf{v}'''], \mathbb{E}[\mathsf{v}'''']) \in \mathsf{O}(\underline{\mathsf{W}}'')_\square$. We still have $(\underline{\mathsf{W}}'', \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau_2]\!]_\square$ by public world monotonicity, so that the result follows in combination with $(\underline{\mathsf{W}}'', \mathbf{v}''', \mathsf{v}'''') \in \mathcal{V}[\![\tau_2]\!]_\square$.

- $\tau = \tau_1 \times \tau_2$: We have (by definition) that

$$\mathsf{protect}_{\tau_1 \times \tau_2} = \lambda \mathsf{y}.\ \langle \mathsf{protect}_{\tau_1}\ \mathsf{y}.1, \mathsf{protect}_{\tau_2}\ \mathsf{y}.2 \rangle$$

and

$$\mathsf{confine}_{\tau_1 \times \tau_2} = \lambda \mathsf{y}.\ \langle \mathsf{confine}_{\tau_1}\ \mathsf{y}.1, \mathsf{confine}_{\tau_2}\ \mathsf{y}.2 \rangle.$$

We do the proof for $\mathsf{protect}_{\tau_1 \times \tau_2}$, the proof for $\mathsf{confine}_{\tau_1 \times \tau_2}$ is symmetric.

We know from $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \times \tau_2]\!]_\square$ that $\mathbf{v} = \langle \mathbf{v_1}, \mathbf{v_2} \rangle$ and $\mathsf{v} = \langle \mathsf{v}_1, \mathsf{v}_2 \rangle$ for some $\mathbf{v_1}, \mathbf{v_2}, \mathsf{v}_1, \mathsf{v}_2$ and that $(\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \triangleright \mathcal{V}[\![\tau_1]\!]_\square$ and $(\underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \triangleright \mathcal{V}[\![\tau_2]\!]_\square$. We also have that $\mathsf{v} \in \mathsf{oftype}(\tau_1 \times \tau_2)$, which implies that $\mathsf{v}_1 \in \mathsf{oftype}(\tau_1)$ and $\mathsf{v}_2 \in \mathsf{oftype}(\tau_2)$.

If $\mathsf{lev}(\underline{\mathsf{W}}) = 0$, then we use Lemma 32 to obtain v_1' and v_2' such that for any $\mathbb{E}$

$$\mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}.2 \rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{protect}_{\tau_2}\ \mathsf{v}.2 \rangle]$$

and

$$\mathbb{E}[\langle \mathsf{v}_1', \mathsf{protect}_{\tau_2}\ \mathsf{v}_2 \rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{v}_2' \rangle],$$

and $\mathsf{v}_1' \in \mathsf{oftype}(\tau_1)$ and $\mathsf{v}_2' \in \mathsf{oftype}(\tau_2)$. We then have for any $\mathbb{E}$ that

$$\begin{aligned}\mathbb{E}[\mathsf{protect}_{\tau_1 \times \tau_2}\ \mathsf{v}] &\hookrightarrow \mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}.1, \mathsf{protect}_{\tau_2}\ \mathsf{v}.2 \rangle] \hookrightarrow \\ &\mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}.2 \rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{protect}_{\tau_2}\ \mathsf{v}.2 \rangle] \hookrightarrow \\ &\qquad \mathbb{E}[\langle \mathsf{v}_1', \mathsf{protect}_{\tau_2}\ \mathsf{v}_2 \rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{v}_2' \rangle],\end{aligned}$$

We then also have that $(\underline{\mathsf{W}}, \langle \mathbf{v_1}, \mathbf{v_2}\rangle, \langle \mathsf{v}_1', \mathsf{v}_2'\rangle) \in \mathcal{V}[\![\tau_1 \times \tau_2]\!]_\square$ by definition and by the fact that $(\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1')$ must be in $\triangleright \mathcal{V}[\![\tau_1]\!]$ because $\mathsf{lev}(\underline{\mathsf{W}}) = 0$ and similarly $(\underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2') \in \triangleright \mathcal{V}[\![\tau_2]\!]$.

If $\mathsf{lev}(\underline{\mathsf{W}}) > 0$, then we have that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\tau_1]\!]_\square$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\tau_2]\!]_\square$. We have for any $\mathbb{E}$ that

$$\begin{aligned}
&\mathbb{E}[\mathsf{protect}_{\tau_1 \times \tau_2}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}.1, \mathsf{protect}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow \\
&\qquad \mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{protect}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow \\
&\qquad\qquad \mathbb{E}[\langle \mathsf{v}_1', \mathsf{protect}_{\tau_2}\ \mathsf{v}_2\rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{v}_2'\rangle],
\end{aligned}$$

where we use the induction hypotheses to obtain v_1' and v_2' such that

$$\mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{protect}_{\tau_2}\ \mathsf{v}.2\rangle]$$

and

$$\mathbb{E}[\langle \mathsf{v}_1', \mathsf{protect}_{\tau_2}\ \mathsf{v}_2\rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{v}_2'\rangle].$$

The induction hypotheses also give us that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1') \in \mathcal{V}[\![\tau_1]\!]_\square$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2') \in \mathcal{V}[\![\tau_2]\!]_\square$.

It remains to prove that $(\underline{\mathsf{W}}, \langle \mathbf{v_1}, \mathbf{v_2}\rangle, \langle \mathsf{v}_1', \mathsf{v}_2'\rangle) \in \mathcal{V}[\![\tau_1 \times \tau_2]\!]_\square$, but this follows easily by definition and by Lemma 17.

- $\tau = \tau_1 \uplus \tau_2$: We have (by definition) that

$$\mathsf{protect}_{\tau_1 \uplus \tau_2} = \lambda \mathsf{y}.\, \mathrm{case}\ \mathsf{y}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x}_1 \mapsto \mathrm{inl}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x}_1) \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathrm{inr}\ (\mathsf{protect}_{\tau_2}\ \mathsf{x}_2)$$

and

$$\mathsf{confine}_{\tau_1 \uplus \tau_2} = \lambda \mathsf{y}.\, \mathrm{case}\ \mathsf{y}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x}_1 \mapsto \mathrm{inl}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}_1) \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathrm{inr}\ (\mathsf{confine}_{\tau_2}\ \mathsf{x}_2).$$

We do the proof for $\mathsf{protect}_{\tau_1 \uplus \tau_2}$, the proof for $\mathsf{confine}_{\tau_1 \uplus \tau_2}$ is symmetric.

We know from $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \uplus \tau_2]\!]_\square$ that either $\mathbf{v} = \mathrm{inl}\ \mathbf{v_1}$ and $\mathsf{v} = \mathrm{inl}\ \mathsf{v}_1$ for some $\mathbf{v_1}, \mathsf{v}_1$ with $(\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \triangleright \mathcal{V}[\![\tau_1]\!]_\square$ or $\mathbf{v} = \mathrm{inr}\ \mathbf{v_2}$ and $\mathsf{v} = \mathrm{inr}\ \mathsf{v}_2$ for some $\mathbf{v_2}, \mathsf{v}_2$ with $(\underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \triangleright \mathcal{V}[\![\tau_2]\!]_\square$. We complete the proof for the first case, the other one is similar.

If $\mathsf{lev}(\underline{\mathsf{W}}) = 0$, then we use Lemma 32 to obtain v_1' and v_2' such that for any $\mathbb{E}$

$$\mathbb{E}[\mathrm{inl}\ (\mathsf{protect}_{\tau_1}\ \mathsf{v}_1)] \hookrightarrow^* \mathbb{E}[\mathrm{inl}\ \mathsf{v}_1'],$$

and $\mathsf{v}_1' \in \mathsf{oftype}(\tau_1)$. We then have for any $\mathbb{E}$ that

$$\begin{aligned}
&\mathbb{E}[\mathsf{protect}_{\tau_1 \uplus \tau_2}\ \mathsf{v}] \hookrightarrow \\
&\mathbb{E}[\mathrm{case}\ \mathsf{v}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x}_1 \mapsto \mathrm{inl}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x}_1) \mid \mathrm{inr}\ \mathsf{x}_2 \mapsto \mathrm{inr}\ (\mathsf{protect}_{\tau_2}\ \mathsf{x}_2)] \hookrightarrow \\
&\qquad\qquad \mathbb{E}[\mathrm{inl}\ (\mathsf{protect}_{\tau_1}\ \mathsf{v}_1)] \hookrightarrow^* \mathbb{E}[\mathrm{inl}\ \mathsf{v}_1'],
\end{aligned}$$

We then also have that $(\underline{\mathsf{W}}, \langle \mathbf{v_1}, \mathbf{v_2}\rangle, \langle \mathsf{v}_1', \mathsf{v}_2'\rangle) \in \mathcal{V}[\![\tau_1 \uplus \tau_2]\!]_\square$ by definition and by the fact that $(\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1')$ must be in $\triangleright \mathcal{V}[\![\tau_1]\!]$ because $\mathsf{lev}(\underline{\mathsf{W}}) = 0$.

If $\mathsf{lev}(\underline{\mathsf{W}}) > 0$, then we have that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\tau_1]\!]_\square$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\tau_2]\!]_\square$. We have for any $\mathbb{E}$ that

$$\begin{aligned}
&\mathbb{E}[\mathsf{protect}_{\tau_1 \uplus \tau_2}\ \mathsf{v}] \hookrightarrow \\
&\mathbb{E}[\text{case } \mathsf{v} \text{ of inl } \mathsf{x}_1 \mapsto \text{inl } (\mathsf{protect}_{\tau_1}\ \mathsf{x}_1) \mid \text{inr } \mathsf{x}_2 \mapsto \text{inr } (\mathsf{protect}_{\tau_2}\ \mathsf{x}_2)] \hookrightarrow \\
&\qquad\qquad\qquad\qquad\qquad\qquad \mathbb{E}[\text{inl } (\mathsf{protect}_{\tau_1}\ \mathsf{v}_1)] \hookrightarrow^* \mathbb{E}[\text{inl } \mathsf{v}'_1],
\end{aligned}$$

where we use the induction hypotheses to obtain v'_1 such that

$$\mathbb{E}[\text{inl } (\mathsf{protect}_{\tau_1}\ \mathsf{v}_1)] \hookrightarrow^* \mathbb{E}[\text{inl } \mathsf{v}'_1]$$

The induction hypotheses also give us that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}'_1) \in \mathcal{V}[\![\tau_1]\!]_\square$. It remains to prove that $(\underline{\mathsf{W}}, \mathbf{v}, \text{inl } \mathsf{v}'_1) \in \mathcal{V}[\![\tau_1 \uplus \tau_2]\!]_\square$, but this follows easily by definition and Lemma 17.

□

Theorem 6 (Protect and confine are semantics preserving)**.** *For any n, if $\mathbf{\Gamma} \vdash \mathbf{t_1} \square_n \mathsf{t}_2 : \tau$ then $\mathbf{\Gamma} \vdash \mathbf{t_1} \square_n \mathsf{protect}_\tau\ \mathsf{t}_2 : \tau$ and $\mathbf{\Gamma} \vdash \mathbf{t_1} \square_n \mathsf{confine}_\tau\ \mathsf{t}_2 : \tau$.*

Proof. We only prove the part about $\mathsf{protect}_\tau$, the result about $\mathsf{confine}_\tau$ is similar.

Take $\underline{\mathsf{W}}$ with $\mathsf{lev}(\underline{\mathsf{W}}) \leq n$, $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_\square$. Then we need to show that $(\underline{\mathsf{W}}, \mathbf{t}\gamma, \mathsf{protect}_\tau\ \mathsf{t}\gamma) \in \mathcal{E}[\![\tau]\!]_\square$. From $\mathbf{\Gamma} \vdash \mathbf{t} \square_n \mathsf{t} : \tau$, we have that $(\underline{\mathsf{W}}, \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\tau]\!]_\square$, so that by Lemma 19, it suffices to prove that for all $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau]\!]_\square$, we have that $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{protect}_\tau\ \mathsf{v}) \in \mathcal{E}[\![\tau]\!]_\square$.

So, take $(\underline{\mathsf{W}}', \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau]\!]_\square$, then we need to show that $(\mathbb{E}[\mathbf{v}], \mathbb{E}[\mathsf{protect}_\tau\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}}')_\square$. From Lemma 33, we get a v' such that $\mathbb{E}[\mathsf{protect}_\tau\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}']$ and $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}') \in \mathcal{V}[\![\tau]\!]_\square$. By Lemma 4, it suffices to prove that $(\mathbb{E}[\mathbf{v}], \mathbb{E}[\mathsf{v}']) \in \mathsf{O}(\underline{\mathsf{W}}')_\square$. This now follows directly from $(\underline{\mathsf{W}}', \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\tau]\!]_\square$ with $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}') \in \mathcal{V}[\![\tau]\!]_\square$. □

5.4 Contextual equivalence reflection

Theorem 7 ($[\![\cdot]\!]^{\lambda^\tau}_{\lambda^u}$ is semantics preserving)**.** *For all $\mathbf{t}$, if $\mathbf{\Gamma} \vdash \mathbf{t} : \tau$ then $\mathbf{\Gamma} \vdash \mathbf{t} \square_n [\![\mathbf{t}]\!]^{\lambda^\tau}_{\lambda^u} : \tau$.*

Proof. By definition, we have that $[\![\mathbf{t}]\!]^{\lambda^\tau}_{\lambda^u} = \mathsf{protect}_\tau\ \mathtt{erase}(\mathbf{t})$. From $\mathbf{\Gamma} \vdash \mathbf{t} : \tau$, we get $\mathbf{\Gamma} \vdash \mathbf{t} \square_n \mathtt{erase}(\mathbf{t}) : \tau$ by Theorem 4. By Theorem 6, we get that $\mathbf{\Gamma} \vdash \mathbf{t} \square_n \mathsf{protect}_\tau\ \mathtt{erase}(\mathbf{t}) : \tau$ as required. □

Theorem 8 ($[\![\cdot]\!]^{\lambda^\tau}_{\lambda^u}$ reflects equivalence)**.** *If $\emptyset \vdash \mathbf{t_1} : \tau$, $\emptyset \vdash \mathbf{t_2} : \tau$ and $\emptyset \vdash \mathsf{protect}_\tau\ \mathtt{erase}(\mathbf{t_1}) \simeq_{ctx} \mathsf{protect}_\tau\ \mathtt{erase}(\mathbf{t_2})$, then $\emptyset \vdash \mathbf{t_1} \simeq_{ctx} \mathbf{t_2} : \tau$.*

Proof. Take $\mathfrak{C}$ so that $\vdash \mathfrak{C} : \emptyset, \tau \to \emptyset, \tau'$. We need to prove that $\mathfrak{C}[\mathbf{t_1}]\Downarrow$ iff $\mathfrak{C}[\mathbf{t_2}]\Downarrow$. By symmetry, it suffices to prove the $\Rightarrow$ direction. So assume that $\mathfrak{C}[\mathbf{t_1}]\Downarrow$, then we need to prove that $\mathfrak{C}[\mathbf{t_2}]\Downarrow$.

Define $\mathfrak{C} \stackrel{\text{def}}{=} \texttt{erase}(\mathfrak{C})$, then Theorem 5 tells us that $\vdash \mathfrak{C} \sqsupseteq_n \mathfrak{C} : \emptyset, \tau \to \emptyset, \tau'$.

From Theorem 7, we get that $\emptyset \vdash \mathbf{t_1} \sqsupseteq_n [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} : \tau$ and $\emptyset \vdash \mathbf{t_2} \sqsupseteq_n [\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u} : \tau$.

By definition of $\vdash \mathfrak{C} \sqsupseteq_n \mathfrak{C} : \emptyset, \tau \to \emptyset, \tau'$, we get that $\emptyset \vdash \mathfrak{C}[\mathbf{t_1}] \sqsupseteq_n \mathfrak{C}[[\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u}] : \tau'$ and $\emptyset \vdash \mathfrak{C}[\mathbf{t_2}] \sqsupseteq_n \mathfrak{C}[[\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}] : \tau'$.

By Lemma 16, $\mathfrak{C}[\mathbf{t_1}]\Downarrow$ and $\emptyset \vdash \mathfrak{C}[\mathbf{t_1}] \sqsupseteq_n \mathfrak{C}[[\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u}] : \tau'$ imply that $\mathfrak{C}[[\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$.

From $\emptyset \vdash [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} \simeq_{ctx} [\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}$ and $\mathfrak{C}[[\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$, we get that $\mathfrak{C}[[\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$, since by Lemma 18, we get $\vdash \mathfrak{C} : \emptyset \to \emptyset$ from $\vdash \mathfrak{C} : \emptyset, \tau \to \emptyset, \tau'$.

By Lemma 16, we now get that $\mathfrak{C}[\mathbf{t_2}]\Downarrow$ from $\emptyset \vdash \mathfrak{C}[\mathbf{t_2}] \sqsupseteq_n \mathfrak{C}[[\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}] : \tau'$ and $\mathfrak{C}[[\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$. $\square$

6 Equivalence preservation and emulation

This section defines UVal (Section 6.1) and clarifies $\mathtt{EmulDV}$ (Section 6.2). Then it introduces upgrade and downgrade (Section 6.3), inject and extract (Section 6.4) and emulate (Section 6.5). Finally it defines the approximate back-translation (Section 6.6) and it proves compiler security (Section 6.7).

6.1 n-approximate UVal

We define a family of λ^{τ} types UVal:

$$\mathsf{UVal}_{\mathsf{0}} \stackrel{\mathsf{def}}{=} \mathtt{Unit}$$

$$\mathsf{UVal}_{\mathsf{n+1}} \stackrel{\mathsf{def}}{=} \mathtt{Unit} \uplus \mathtt{Unit} \uplus \mathtt{Bool} \uplus (\mathsf{UVal}_{\mathsf{n}} \times \mathsf{UVal}_{\mathsf{n}}) \uplus (\mathsf{UVal}_{\mathsf{n}} \to \mathsf{UVal}_{\mathsf{n}}) \uplus (\mathsf{UVal}_{\mathsf{n}} \uplus \mathsf{UVal}_{\mathsf{n}})$$

Note: in $\mathsf{UVal}_{\mathsf{n+1}}$, the first $\mathtt{Unit}$ represents an emulation of an unknown value and the second $\mathtt{Unit}$ represents the emulation of an actual $\mathtt{Unit}$ value. We define the following functions with the obvious implementations:

$$\begin{aligned}
\mathbf{in}_{\mathrm{unk};\mathbf{n}} &: \mathsf{UVal}_{\mathsf{n+1}} \\
\mathbf{in}_{\mathtt{Unit};\mathbf{n}} &: \mathtt{Unit} \to \mathsf{UVal}_{\mathsf{n+1}} \\
\mathbf{in}_{\mathtt{Bool};\mathbf{n}} &: \mathtt{Bool} \to \mathsf{UVal}_{\mathsf{n+1}} \\
\mathbf{in}_{\times;\mathbf{n}} &: (\mathsf{UVal}_{\mathsf{n}} \times \mathsf{UVal}_{\mathsf{n}}) \to \mathsf{UVal}_{\mathsf{n+1}} \\
\mathbf{in}_{\uplus;\mathbf{n}} &: (\mathsf{UVal}_{\mathsf{n}} \uplus \mathsf{UVal}_{\mathsf{n}}) \to \mathsf{UVal}_{\mathsf{n+1}} \\
\mathbf{in}_{\to;\mathbf{n}} &: (\mathsf{UVal}_{\mathsf{n}} \to \mathsf{UVal}_{\mathsf{n}}) \to \mathsf{UVal}_{\mathsf{n+1}}
\end{aligned}$$

We also define a convenience meta-level function for constructing an unknown $\mathsf{UVal}_{\mathsf{n}}$ for an arbitrary n:

$$\begin{aligned}
\mathrm{unk}_{\mathsf{n}} &: \mathsf{UVal}_{\mathsf{n}} \\
\mathrm{unk}_{\mathsf{0}} &\stackrel{\mathsf{def}}{=} \mathtt{unit} \\
\mathrm{unk}_{\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \mathbf{in}_{\mathrm{unk};\mathbf{n}}
\end{aligned}$$

We also define the following functions:

$$\mathrm{omega}_\tau : \tau$$

$$\mathrm{omega}_\tau \overset{\mathsf{def}}{=} \mathrm{fix}_{\mathrm{unit}\to\tau}\ (\lambda\mathbf{x} : \mathrm{unit} \to \tau.\, \mathbf{x})\ \mathrm{unit}$$

$$\mathtt{case}_{\mathtt{Unit};\mathsf{n}} : \mathrm{UVal}_{\mathsf{n}+1} \to \mathtt{Unit}$$

$$\mathtt{case}_{\mathtt{Bool};\mathsf{n}} : \mathrm{UVal}_{\mathsf{n}+1} \to \mathtt{Bool}$$

$$\mathtt{case}_{\times;\mathsf{n}} : \mathrm{UVal}_{\mathsf{n}+1} \to (\mathrm{UVal}_{\mathsf{n}} \times \mathrm{UVal}_{\mathsf{n}})$$

$$\mathtt{case}_{\uplus;\mathsf{n}} : \mathrm{UVal}_{\mathsf{n}+1} \to (\mathrm{UVal}_{\mathsf{n}} \uplus \mathrm{UVal}_{\mathsf{n}})$$

$$\mathtt{case}_{\to;\mathsf{n}} : \mathrm{UVal}_{\mathsf{n}+1} \to \mathrm{UVal}_{\mathsf{n}} \to \mathrm{UVal}_{\mathsf{n}}$$

$$\mathtt{case}_{\mathtt{Unit};\mathsf{n}} \overset{\mathsf{def}}{=} \lambda\mathbf{x} : \mathrm{UVal}_{\mathsf{n}+1}.\, \mathrm{case}\ \mathbf{x}\ \mathrm{of}\ \{\mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{\mathtt{Unit}}\}$$

$$\mathtt{case}_{\mathtt{Bool};\mathsf{n}} \overset{\mathsf{def}}{=} \lambda\mathbf{x} : \mathrm{UVal}_{\mathsf{n}+1}.\, \mathrm{case}\ \mathbf{x}\ \mathrm{of}\ \{\mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{\mathtt{Bool}}\}$$

$$\mathtt{case}_{\times;\mathsf{n}} \overset{\mathsf{def}}{=} \lambda\mathbf{x} : \mathrm{UVal}_{\mathsf{n}+1}.\, \mathrm{case}\ \mathbf{x}\ \mathrm{of}\ \{\mathbf{in}_{\times;\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{(\mathrm{UVal}_{\mathsf{n}} \times \mathrm{UVal}_{\mathsf{n}})}\}$$

$$\mathtt{case}_{\uplus;\mathsf{n}} \overset{\mathsf{def}}{=} \lambda\mathbf{x} : \mathrm{UVal}_{\mathsf{n}+1}.\, \mathrm{case}\ \mathbf{x}\ \mathrm{of}\ \{\mathbf{in}_{\uplus;\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{(\mathrm{UVal}_{\mathsf{n}} \uplus \mathrm{UVal}_{\mathsf{n}})}\}$$

$$\mathtt{case}_{\to;\mathsf{n}} \overset{\mathsf{def}}{=} \lambda\mathbf{x} : \mathrm{UVal}_{\mathsf{n}+1}.\, \lambda\mathbf{y} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{case}\ \mathbf{x}\ \mathrm{of}\ \{\mathbf{in}_{\to;\mathbf{n}}\ \mathbf{z} \mapsto \mathbf{z}\ \mathbf{y};\ \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_{\mathsf{n}}}\}$$

Lemma 34 (omega diverges)**.** *For any τ and any evaluation context $\mathbb{E}$, $\mathbb{E}[\mathrm{omega}_\tau]\Uparrow$, i.e. it diverges.*

Proof. We have the following:

$$\begin{aligned}\mathbb{E}[\mathrm{omega}_\tau] = \mathbb{E}[\mathrm{fix}_{\mathrm{unit}\to\tau}\ (\lambda\mathbf{x} : \mathrm{unit} \to \tau.\, \mathbf{x})\ \mathrm{unit}] \hookrightarrow \\ \mathbb{E}[(\lambda\mathbf{y} : \mathrm{unit}.\, \mathrm{fix}_{\mathrm{unit}\to\tau}\ (\lambda\mathbf{x} : \mathrm{unit}.\, \mathbf{x})\ \mathbf{y})\ \mathrm{unit}] \hookrightarrow \\ \mathbb{E}[\mathrm{fix}_{\mathrm{unit}\to\tau}\ (\lambda\mathbf{x} : \mathrm{unit}.\, \mathbf{x})\ \mathrm{unit}] = \mathbb{E}[\mathrm{omega}_\tau]\end{aligned}$$

In summary, $\mathbb{E}[\mathrm{omega}_\tau] \hookrightarrow^{\mathbf{2}} \mathbb{E}[\mathrm{omega}_\tau]$, so that it must diverge. □

6.2 EmulDV specification

We use an indexed definition of $\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}$ that takes into account the fact that we have a step-indexed UVal now. In fact, we need two indices n and p. The first index n is a non-negative number which determines the type of the λ^τ term, i.e. if $(\underline{\mathrm{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]$, then we must have that $\emptyset \vdash \mathbf{v} : \mathrm{UVal}_{\mathsf{n}}$. The index p must either be `precise` or `imprecise` and determines the level up to which the term is accurate. If p is `imprecise`, the term may contain $\mathbf{in}_{\mathrm{unk};\mathbf{n}}$ values corresponding to arbitrary λ^{u} values. However, if p is `precise`, it must not contain $\mathbf{in}_{\mathrm{unk};\mathbf{n}}$, at least up to the level determined by the amount of steps in the world.

6.3 Upgrade/downgrade

We define the following functions:

$$\mathrm{downgrade}_{\mathsf{n;d}} : \mathrm{UVal}_{\mathsf{n+d}} \to \mathrm{UVal}_{\mathsf{n}}$$

$$\mathrm{downgrade}_{\mathsf{0;d}} \stackrel{\mathsf{def}}{=} \lambda \mathbf{v} : \mathrm{UVal}_{\mathbf{d}}.\, \mathrm{unit}$$

$$\mathrm{downgrade}_{\mathsf{n+1;d}} \stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+d+1}}.\, \mathsf{case}\ \mathbf{x}\ \mathsf{of} \begin{cases} \mathbf{in}_{\mathrm{unk};\mathbf{n+d}} \mapsto \mathbf{in}_{\mathrm{unk};\mathbf{n}}; \\ \mathbf{in}_{\mathtt{Unit};\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ y; \\ \mathbf{in}_{\mathtt{Bool};\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ y; \\ \mathbf{in}_{\times;\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\times;\mathbf{n}}\ \langle \mathrm{downgrade}_{\mathsf{n;d}}\ y.1, \mathrm{downgrade}_{\mathsf{n;d}}\ y.2 \rangle; \\ \mathbf{in}_{\uplus;\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\uplus;\mathbf{n}}\ \mathsf{case}\ y\ \mathsf{of}\ \mathrm{inl}\ x \mapsto \mathrm{inl}\ (\mathrm{downgrade}_{\mathsf{n;d}}\ x); \mathrm{inr}\ x \mapsto \mathrm{inr}\ (\mathrm{downgrade}_{\mathsf{n;d}}\ x) \\ \mathbf{in}_{\to;\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\to;\mathbf{n}}\ (\lambda z : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{downgrade}_{\mathsf{n;d}}\ (y\ (\mathrm{upgrade}_{\mathsf{n;d}}\ z))) \end{cases}$$

$$\mathrm{upgrade}_{\mathsf{n;d}} : \mathrm{UVal}_{\mathsf{n}} \to \mathrm{UVal}_{\mathsf{n+d}}$$

$$\mathrm{upgrade}_{\mathsf{0;d}} \stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathbf{0}}.\, \mathrm{unk}_{\mathbf{d}}$$

$$\mathrm{upgrade}_{\mathsf{n+1;d}} \stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+1}}.\, \mathsf{case}\ \mathbf{x}\ \mathsf{of} \begin{cases} \mathbf{in}_{\mathrm{unk};\mathbf{n}} \mapsto \mathbf{in}_{\mathrm{unk};\mathbf{n+d}}; \\ \mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ y \mapsto \mathbf{in}_{\mathtt{Unit};\mathbf{n+d}}\ y; \\ \mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ y \mapsto \mathbf{in}_{\mathtt{Bool};\mathbf{n+d}}\ y; \\ \mathbf{in}_{\times;\mathbf{n}}\ y \mapsto \mathbf{in}_{\times;\mathbf{n+d}}\ \langle \mathrm{upgrade}_{\mathsf{n;d}}\ y.1, \mathrm{upgrade}_{\mathsf{n;d}}\ y.2 \rangle; \\ \mathbf{in}_{\uplus;\mathbf{n}}\ y \mapsto \mathbf{in}_{\uplus;\mathbf{n+d}}\ \mathsf{case}\ y\ \mathsf{of}\ \mathrm{inl}\ x \mapsto \mathrm{inl}\ (\mathrm{upgrade}_{\mathsf{n;d}}\ x); \mathrm{inr}\ x \mapsto \mathrm{inr}\ (\mathrm{upgrade}_{\mathsf{n;d}}\ x) \\ \mathbf{in}_{\to;\mathbf{n}}\ y \mapsto \mathbf{in}_{\to;\mathbf{n+d}}\ (\lambda z : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{upgrade}_{\mathsf{n;d}}\ (y\ (\mathrm{downgrade}_{\mathsf{n;d}}\ z))) \end{cases}$$

Lemma 35 (Upgrade and downgrade are well-typed)**.** *For all n, d, $\mathrm{upgrade}_{\mathsf{n;d}} : \mathrm{UVal}_{\mathsf{n}} \to \mathrm{UVal}_{\mathsf{n+d}}$ and $\mathrm{downgrade}_{\mathsf{n;d}} : \mathrm{UVal}_{\mathsf{n+d}} \to \mathrm{UVal}_{\mathsf{n}}$.*

Proof. Easily verified. □

Lemma 36 (Upgrade and downgrade reduce)**.** *If $\emptyset \vdash \mathbf{v} : \mathrm{UVal}_{\mathsf{n+d}}$, then for any $\mathbb{E}$, $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ for some $\mathbf{v}'$.*

If $\emptyset \vdash \mathbf{v} : \mathrm{UVal}_{\mathsf{n}}$, then for any $\mathbb{E}$, $\mathbb{E}[\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ for some $\mathbf{v}'$.

Proof. Take $\emptyset \vdash \mathbf{v} : \mathrm{UVal}_{\mathsf{n+d}}$ and an arbitrary $\mathbb{E}$. We prove that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ by induction on the structure of $\mathbf{v}$.

If $n = 0$, then we have that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v}] = \mathbb{E}[(\lambda \mathbf{x} : \mathrm{UVal}_{\mathbf{d}}.\, \mathrm{unit})\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathrm{unit}]$.

For $n+1$, we have by a standard canonicity lemma, that one of the following holds:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};\mathbf{n+d}}$. In this case, we have that

$$\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\mathrm{unk};\mathbf{n}}]$$

- $\mathbf{v} = \mathbf{in}_{\mathtt{Unit};\mathbf{n+d}}\ \mathbf{v'}$. In this case, we have that

$$\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ \mathbf{v'}]$$

- $\mathbf{v} = \mathbf{in}_{\mathtt{Bool};\mathbf{n+d}}\ \mathbf{v'}$. In this case, we have that

$$\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ \mathbf{v'}]$$

- $\mathbf{v} = \mathbf{in}_{\times;\mathbf{n+d}}\ \langle \mathbf{v_1}, \mathbf{v_2} \rangle$ with $\mathbf{v_1} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n+d}})$ and $\mathbf{v_2} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n+d}})$. In this case, we have that

$$\begin{aligned}\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.1}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow \\ \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_1}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow^* \\ \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v'_1}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow \\ \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v'_1}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_2}\rangle)] \hookrightarrow^* \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v'_1}, \mathbf{v'_2}\rangle)]\end{aligned}$$

  where we use the fact that by induction $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v'_1}]$ and $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_2}] \hookrightarrow^* \mathbb{E}[\mathbf{v'_2}]$ for some $\mathbf{v'_1}, \mathbf{v'_2}$ for any $\mathbb{E}$.

- $\mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n+d}}(\mathrm{inl}\ \mathbf{v_1})$ with $\mathbf{v_1} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n+d}})$ or $\mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n+d}}(\mathrm{inr}\ \mathbf{v_2})$ with $\mathbf{v_2} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n+d}})$. We only treat the first case, the other is similar. We then have that

$$\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n}}(\mathrm{inl}\ (\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_1}))] \hookrightarrow^* \mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n}}(\mathrm{inl}\ \mathbf{v'_1})]$$

  where we use the fact that by induction $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v'_1}]$ for some $\mathbf{v'_1}$ for any $\mathbb{E}$.

- $\mathbf{v} = \mathbf{in}_{\to;\mathbf{n+d}}(\mathbf{v'})$ with $\mathbf{v} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n+d}} \to \mathrm{UVal}_{\mathsf{n+d}})$. We then have that

$$\begin{aligned}&\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \\ &\qquad \mathbb{E}[\mathbf{in}_{\to;\mathbf{n}}(\lambda \mathbf{z} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{downgrade}_{\mathsf{n;d}}\ (\mathbf{y}\ (\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{z})))],\end{aligned}$$

  which is clearly a value.

Now take $\mathbf{v} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n}})$. We prove that $\mathbb{E}[\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v'}]$ by induction on the structure of $\mathbf{v}$.

If $n = 0$, then we have that $\mathbb{E}[\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v}] = \mathbb{E}[(\lambda \mathbf{x} : \mathrm{UVal}_{\mathbf{0}}.\, \mathrm{unk}_{\mathbf{d}})\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathrm{unk}_{\mathbf{d}}]$, and we know that unk_d is always a value.

For $n+1$, we have by a standard canonicity lemma, that one of the following holds:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};\mathbf{n}}$. In this case, we have that

$$\mathbb{E}[\mathrm{upgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\mathrm{unk};\mathbf{n+d}}]$$

- $\mathbf{v} = \mathbf{in}_{\texttt{Unit};\mathbf{n}}(\mathbf{v}')$. In this case, we have that

$$\mathbb{E}[\mathrm{upgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\texttt{Unit};\mathbf{n+d}}(\mathbf{v}')]$$

- $\mathbf{v} = \mathbf{in}_{\texttt{Bool};\mathbf{n}}(\mathbf{v}')$. In this case, we have that

$$\mathbb{E}[\mathrm{upgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\texttt{Bool};\mathbf{n+d}}(\mathbf{v}')]$$

- $\mathbf{v} = \mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v_1}, \mathbf{v_2}\rangle)$ with $\mathbf{v_1} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n}})$ and $\mathbf{v_2} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n}})$. In this case, we have that

$$\begin{aligned}\mathbb{E}[\mathrm{upgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v.1}, \mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow \\ \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v_1}, \mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow^* \\ \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v}'_{\mathbf{1}}, \mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow \\ \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v}'_{\mathbf{1}}, \mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v_2}\rangle)] \hookrightarrow^* \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v}'_{\mathbf{1}}, \mathbf{v}'_{\mathbf{2}}\rangle)]\end{aligned}$$

  where we use the fact that by induction $\mathbb{E}[\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v}'_{\mathbf{1}}]$ and $\mathbb{E}[\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v_2}] \hookrightarrow^* \mathbb{E}[\mathbf{v}'_{\mathbf{2}}]$ for some $\mathbf{v}'_{\mathbf{1}}, \mathbf{v}'_{\mathbf{2}}$ for any $\mathbb{E}$.

- $\mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathrm{inl}\ \mathbf{v_1})$ with $\mathbf{v_1} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n}})$ or $\mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathrm{inr}\ \mathbf{v_2})$ with $\mathbf{v_2} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n}})$. We only treat the first case, the other is similar. We then have that

$$\begin{aligned}\mathbb{E}[\mathrm{upgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n+d}}(\mathrm{inl}\ (\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v.1}))] \hookrightarrow \\ \mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n+d}}(\mathrm{inl}\ (\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v_1}))] \hookrightarrow^* \mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n+d}}(\mathrm{inl}\ \mathbf{v}'_{\mathbf{1}})]\end{aligned}$$

  where we use the fact that by induction $\mathbb{E}[\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v}'_{\mathbf{1}}]$ for some $\mathbf{v}'_{\mathbf{1}}$ for any $\mathbb{E}$.

- $\mathbf{v} = \mathbf{in}_{\to;\mathbf{n}}(\mathbf{v}')$ with $\mathbf{v} \in \mathbf{oftype}(\mathrm{UVal}_{\mathsf{n}} \to \mathrm{UVal}_{\mathsf{n}})$. We then have that

$$\begin{aligned}\mathbb{E}[\mathrm{upgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow \\ \mathbb{E}[\mathbf{in}_{\to;\mathbf{n+d}}(\lambda \mathbf{z} : \mathrm{UVal}_{\mathsf{n+d}}.\ \mathrm{upgrade}_{\mathsf{n;d}}\ (\mathbf{y}\ (\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{z})))],\end{aligned}$$

  which is clearly a value.

$\square$

Lemma 37 (Related upgraded terms reduce and they are still related)**.** *If ($\mathsf{lev}(\underline{\mathsf{W}}) < n$ and $p = \texttt{precise}$) or ($\square =\lesssim$ and $p = \texttt{imprecise}$), and if $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$, then there exists a $\mathbf{v}'$ such that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ for any $\mathbb{E}$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$.*

If ($\mathsf{lev}(\underline{\mathsf{W}}) < n$ and $p = \texttt{precise}$) or ($\square =\lesssim$ and $p = \texttt{imprecise}$), then if $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$, then there exists a $\mathbf{v}'$ such that $\mathbb{E}[\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ for any $\mathbb{E}$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$.

Proof. We prove both results simultaneously by induction on n.

If $n = 0$, then take $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$. We have that $\mathrm{downgrade}_{\mathsf{0;d}} = \lambda \mathbf{v} : \mathrm{UVal}_{\mathbf{d}}.\, \mathrm{unit}$, so that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{0;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathrm{unit}]$ for any $\mathbb{E}$. By definition of $\mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{0;p}}]\!]$, we have that $(\underline{\mathsf{W}}, \mathrm{unit}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{0;p}}]\!]_{\square}$.

Still if $n = 0$, take $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$. We have that $\mathrm{upgrade}_{\mathsf{0;d}} = \lambda \mathbf{x} : \mathrm{UVal}_{\mathbf{0}}.\, \mathrm{unk}_{\mathbf{d}}$, so that $\mathbb{E}[\mathrm{upgrade}_{\mathsf{0;d}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathrm{unk}_{\mathbf{d}}]$ for any $\mathbb{E}$. If $p = \mathtt{imprecise}$, then we have by definition that $(\underline{\mathsf{W}}, \mathrm{unk}_{\mathbf{d}}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{d;p}}]\!]_{\square}$. $\mathsf{lev}(\underline{\mathsf{W}}) < n = 0$ is not possible.

So now let us prove the results for $n + 1$. We have that

$$\mathrm{downgrade}_{\mathsf{n+1;d}} \stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+d+1}}.\, \mathrm{case}\ \mathbf{x}\ \mathrm{of}$$
$$\begin{cases} \mathbf{in}_{\mathrm{unk};\mathbf{n+d}} \mapsto \mathbf{in}_{\mathrm{unk};\mathbf{n}}; \\ \mathbf{in}_{\mathtt{Unit};\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ y; \\ \mathbf{in}_{\mathtt{Bool};\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ y; \\ \mathbf{in}_{\times;\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\times;\mathbf{n}}\ \langle \mathrm{downgrade}_{\mathsf{n;d}}\ y.1, \mathrm{downgrade}_{\mathsf{n;d}}\ y.2\rangle; \\ \mathbf{in}_{\uplus;\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\uplus;\mathbf{n}}\ \mathrm{case}\ y\ \mathrm{of}\ \mathrm{inl}\ x \mapsto \mathrm{inl}\ (\mathrm{downgrade}_{\mathsf{n;d}}\ x); \mathrm{inr}\ x \mapsto \mathrm{inr}\ (\mathrm{downgrade}_{\mathsf{n;d}}\ x) \\ \mathbf{in}_{\to;\mathbf{n+d}}\ y \mapsto \mathbf{in}_{\to;\mathbf{n}}\ (\lambda z : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{downgrade}_{\mathsf{n;d}}\ (y\ (\mathrm{upgrade}_{\mathsf{n;d}}\ z))) \end{cases}$$

and

$$\mathrm{upgrade}_{\mathsf{n+1;d}} \stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+1}}.\, \mathrm{case}\ \mathbf{x}\ \mathrm{of}$$
$$\begin{cases} \mathbf{in}_{\mathrm{unk};\mathbf{n}} \mapsto \mathbf{in}_{\mathrm{unk};\mathbf{n+d}}; \\ \mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ y \mapsto \mathbf{in}_{\mathtt{Unit};\mathbf{n+d}}\ y; \\ \mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ y \mapsto \mathbf{in}_{\mathtt{Bool};\mathbf{n+d}}\ y; \\ \mathbf{in}_{\times;\mathbf{n}}\ y \mapsto \mathbf{in}_{\times;\mathbf{n+d}}\ \langle \mathrm{upgrade}_{\mathsf{n;d}}\ y.1, \mathrm{upgrade}_{\mathsf{n;d}}\ y.2\rangle; \\ \mathbf{in}_{\uplus;\mathbf{n}}\ y \mapsto \mathbf{in}_{\uplus;\mathbf{n+d}}\ \mathrm{case}\ y\ \mathrm{of}\ \mathrm{inl}\ x \mapsto \mathrm{inl}\ (\mathrm{upgrade}_{\mathsf{n;d}}\ x); \mathrm{inr}\ x \mapsto \mathrm{inr}\ (\mathrm{upgrade}_{\mathsf{n;d}}\ x) \\ \mathbf{in}_{\to;\mathbf{n}}\ y \mapsto \mathbf{in}_{\to;\mathbf{n+d}}\ (\lambda z : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{upgrade}_{\mathsf{n;d}}\ (y\ (\mathrm{downgrade}_{\mathsf{n;d}}\ z))) \end{cases}$$

If $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d+1;p}}]\!]_{\square}$, then we have by definition that one of the following must hold:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};\mathbf{n+d}}$ and $p = \mathtt{imprecise}$. We know that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{in}_{\mathrm{unk};\mathbf{n+d}}] \hookrightarrow^* \mathbb{E}[\mathbf{in}_{\mathrm{unk};\mathbf{n}}]$. It follows directly that $(\underline{\mathsf{W}}, \mathbf{in}_{\mathrm{unk};\mathbf{n}}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+1;p}}]\!]_{\square}$, since $p = \mathtt{imprecise}$.

- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\mathcal{B};\mathbf{n+d}}(\mathbf{v}')$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$. In this case, we have for any $\mathbb{E}$ that
$$\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{in}_{\mathcal{B};\mathbf{n}}(\mathbf{v}')],$$
for any $\mathbb{E}$ and it remains to prove that $(\underline{\mathsf{W}}, \mathbf{in}_{\mathcal{B};\mathbf{n}}(\mathbf{v}'), \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+1;p}}]\!]_{\square}$, but this follows immediately by definition of $\mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+1;p}}]\!]_{\square}$.

- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\times;\mathbf{n+d}}(\mathbf{v}')$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}} \times \mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$. The latter implies that $\mathbf{v}' = \langle \mathbf{v_1}, \mathbf{v_2}\rangle$ and $\mathsf{v} = \langle \mathsf{v_1}, \mathsf{v_2}\rangle$ for $\mathbf{v_1}, \mathbf{v_2}, \mathsf{v_1}, \mathsf{v_2}$ with $(\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v_1}) \in \triangleright\, \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$ and $(\underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v_2}) \in \triangleright\, \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$.

If $\mathsf{lev}(\underline{\mathsf{W}}) = 0$, then we know by Lemma 17 that $\mathbf{v}' \in \mathbf{oftype}(\mathtt{EmulDV}_{\mathsf{n+d;p}} \times \mathtt{EmulDV}_{\mathsf{n+d;p}})$, from which it follows that $\mathbf{v_1} \in \mathbf{oftype}(\mathtt{EmulDV}_{\mathsf{n+d;p}})$ and $\mathbf{v_2} \in \mathbf{oftype}(\mathtt{EmulDV}_{\mathsf{n+d;p}})$, i.e. $\emptyset \vdash \mathbf{v_1} : \mathrm{UVal}_{\mathsf{n+d}}$ and $\emptyset \vdash \mathbf{v_2} : \mathrm{UVal}_{\mathsf{n+d}}$. By Lemma 36, we then get $\mathbf{v_1'}, \mathbf{v_2'}$ such that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v_1'}]$ for any $\mathbb{E}$ and $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_2}] \hookrightarrow^* \mathbb{E}[\mathbf{v_2'}]$ for any $\mathbb{E}$. It follows for any $\mathbb{E}$ that

$$
\begin{aligned}
&\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow^* \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.1}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_1}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow^* \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v_1'}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v_1'}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_2}\rangle)] \hookrightarrow^* \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v_1'}, \mathbf{v_2'}\rangle)]
\end{aligned}
$$

and we have that $(\underline{\mathsf{W}}, \mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v_1'}, \mathbf{v_2'}\rangle), \langle \mathsf{v_1}, \mathsf{v_2}\rangle) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+1;p}}]\!]_{\square}$ by definition and by the fact that $\mathsf{lev}(\underline{\mathsf{W}}) = 0$.

If $\mathsf{lev}(\underline{\mathsf{W}}) > 0$, then we have that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v_1}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v_2}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$.

By induction, we have that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_1}] \hookrightarrow^* \mathbf{v_1'}$ and $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_2}] \hookrightarrow^* \mathbf{v_2'}$ for some $\mathbf{v_1'}, \mathbf{v_2'}$ with $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1'}, \mathsf{v_1}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v_2'}, \mathsf{v_2}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$.

We then also have for any $\mathbb{E}$ that

$$
\begin{aligned}
&\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow^* \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.1}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_1}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow^* \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v_1'}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v.2}\rangle)] \hookrightarrow \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v_1'}, \mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v_2}\rangle)] \hookrightarrow^* \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v_1'}, \mathbf{v_2'}\rangle)]
\end{aligned}
$$

and we have that $(\underline{\mathsf{W}}, \mathbf{in}_{\times;\mathbf{n}}(\langle \mathbf{v_1'}, \mathbf{v_2'}\rangle), \langle \mathsf{v_1}, \mathsf{v_2}\rangle) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$ by definition and by the facts that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1'}, \mathsf{v_1}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v_2'}, \mathsf{v_2}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$.

- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n+d}}(\mathbf{v}')$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}} \uplus \mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$. Similar to the previous case.

- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\to;\mathbf{n+d}}(\mathbf{v}')$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}} \to \mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$. We have that

$$
\begin{aligned}
&\mathbb{E}[\mathrm{downgrade}_{\mathsf{n+1;d}}\ \mathbf{v}] \hookrightarrow^* \\
&\qquad \mathbb{E}[\mathbf{in}_{\to;\mathbf{n}}\ (\lambda \mathbf{z} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{downgrade}_{\mathsf{n;d}}\ (\mathbf{v}'\ (\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{z})))]
\end{aligned}
$$

It remains to show that

$$(\underline{W}, \lambda \mathbf{z} : \mathrm{UVal}_{n}.\, \mathrm{downgrade}_{n;d}\ (\mathbf{v}'\ (\mathrm{upgrade}_{n;d}\ \mathbf{z})), \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \to \mathtt{EmulDV}_{n;p}]\!]_{\square}.$$

From $(\underline{W}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d;p} \to \mathtt{EmulDV}_{n+d;p}]\!]_{\square}$, we have that $\mathbf{v}' = \lambda \mathbf{x} : \mathrm{UVal}_{n+d}.\, \mathbf{t}$ and $\mathsf{v} = \lambda \mathsf{x}.\, \mathsf{t}$ for some $\mathbf{t}$, t.

We need to prove that $\lambda \mathbf{z} : \mathrm{UVal}_{n}.\, \mathrm{downgrade}_{n;d}\ (\mathbf{v}'\ (\mathrm{upgrade}_{n;d}\ \mathbf{z}))$ in $\mathbf{oftype}(\mathtt{EmulDV}_{n;p} \to \mathtt{EmulDV}_{n;p})$, which follows from Lemma 35 and rule λ^{τ}-Type-fun.

Now take $\underline{W}' \sqsupset \underline{W}$, $(\underline{W}', \mathbf{v}'', \mathsf{v}'') \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$, then we need to show that

$$(\underline{W}', \mathrm{downgrade}_{n;d}\ (\mathbf{v}'\ (\mathrm{upgrade}_{n;d}\ \mathbf{v}'')), \mathsf{t}[\mathsf{v}''/\mathsf{x}]) \in \mathcal{E}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}.$$

By induction, we get a $\mathbf{v}'''$ such that $\mathbb{E}[\mathrm{upgrade}_{n;d}\ \mathbf{v}''] \hookrightarrow^{*} \mathbb{E}[\mathbf{v}''']$ for any $\mathbb{E}$ and $(\underline{W}', \mathbf{v}''', \mathsf{v}'') \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d;p}]\!]_{\square}$. We also have that $\mathbb{E}[\mathbf{v}'\ \mathbf{v}'''] \hookrightarrow \mathbb{E}[\mathbf{t}[\mathbf{v}'''/\mathbf{x}]]$. By Lemma 8, it suffices to prove that

$$(\underline{W}', \mathrm{downgrade}_{n;d}\ (\mathbf{t}[\mathbf{v}'''/\mathbf{x}]), \mathsf{t}[\mathsf{v}''/\mathsf{x}]) \in \mathcal{E}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}.$$

Since we know that $(\underline{W}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d;p} \to \mathtt{EmulDV}_{n+d;p}]\!]_{\square}$, $\underline{W}' \sqsupset \underline{W}$ and $(\underline{W}', \mathbf{v}''', \mathsf{v}'') \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d;p}]\!]_{\square}$, it follows that

$$(\underline{W}', \mathbf{t}[\mathbf{v}'''/\mathbf{x}], \mathsf{t}[\mathsf{v}''/\mathsf{x}]) \in \mathcal{E}[\![\mathtt{EmulDV}_{n+d;p}]\!]_{\square}.$$

By Lemma 19, it now suffices to show that for all $\underline{W}'' \sqsupseteq \underline{W}'$, $(\underline{W}'', \mathbf{v_4}, \mathsf{v}_4) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d;p}]\!]_{\square}$, we have that $(\underline{W}'', \mathrm{downgrade}_{n;d}\ \mathbf{v_4}, \mathsf{v}_4) \in \mathcal{E}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$. By induction, we get that $\mathbb{E}[\mathrm{downgrade}_{n;d}\ \mathbf{v_4}] \hookrightarrow^{*} \mathbb{E}[\mathbf{v_5}]$ for any $\mathbb{E}$, for some $\mathbf{v_5}$ with $(\underline{W}'', \mathbf{v_5}, \mathsf{v}_4) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d;p}]\!]_{\square}$. By Lemma 8, it suffices to prove that $(\underline{W}'', \mathbf{v_5}, \mathsf{v}_4) \in \mathcal{E}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$, but this follows directly using Lemma 10.

Now take $(\underline{W}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+1;p}]\!]_{\square}$. then we have by definition that one of the following must hold:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};\mathbf{n}}$ and $p = \mathtt{imprecise}$. We have that $\mathbb{E}[\mathrm{upgrade}_{n+1;d}\ \mathbf{v}] \hookrightarrow^{*} \mathbb{E}[\mathbf{in}_{\mathrm{unk};\mathbf{n+d}}]$ for any $\mathbb{E}$. It follows directly that $(\underline{W}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d+1;p}]\!]_{\square}$, since $p = \mathtt{imprecise}$.
- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\mathcal{B};\mathbf{n}}(\mathbf{v}')$ and $(\underline{W}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$. In this case, we have for any $\mathbb{E}$ that
$$\mathbb{E}[\mathrm{upgrade}_{n+1;d}\ \mathbf{v}] \hookrightarrow^{*} \mathbb{E}[\mathbf{in}_{\mathcal{B};\mathbf{n+d}}(\mathbf{v}')],$$
for any $\mathbb{E}$ and it remains to prove that $(\underline{W}, \mathbf{in}_{\mathcal{B};\mathbf{n+d}}(\mathbf{v}'), \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d+1;p}]\!]_{\square}$, but this follows immediately by definition of $\mathcal{V}[\![\mathtt{EmulDV}_{n+d+1;p}]\!]_{\square}$.

- $\exists \mathbf{v'}.\, \mathbf{v} = \mathbf{in}_{\times;\mathbf{n}}(\mathbf{v'})$ and $(\underline{W}, \mathbf{v'}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \times \mathtt{EmulDV}_{n;p}]\!]_{\square}$. The latter implies that $\mathbf{v'} = \langle \mathbf{v_1}, \mathbf{v_2} \rangle$ and $\mathsf{v} = \langle \mathsf{v}_1, \mathsf{v}_2 \rangle$ for $\mathbf{v_1}, \mathbf{v_2}, \mathsf{v}_1, \mathsf{v}_2$ with $(\underline{W}, \mathbf{v_1}, \mathsf{v}_1) \in \triangleright \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$ and $(\underline{W}, \mathbf{v_2}, \mathsf{v}_2) \in \triangleright \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$.

  If $\mathsf{lev}(\underline{W}) = 0$, then we know by Lemma 17 that $\mathbf{v'} \in \mathbf{oftype}(\mathtt{EmulDV}_{n;p} \times \mathtt{EmulDV}_{n;p})$, from which it follows that $\mathbf{v_1} \in \mathbf{oftype}(\mathtt{EmulDV}_{n;p})$ and $\mathbf{v_2} \in \mathbf{oftype}(\mathtt{EmulDV}_{n;p})$, which imply $\emptyset \vdash \mathbf{v_1} : \mathrm{UVal}_n$ and $\emptyset \vdash \mathbf{v_2} : \mathrm{UVal}_n$. By Lemma 36, we then get $\mathbf{v'_1}, \mathbf{v'_2}$ such that $\mathbb{E}[\mathrm{upgrade}_{n;d}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v'_1}]$ and $\mathbb{E}[\mathrm{upgrade}_{n;d}\ \mathbf{v_2}] \hookrightarrow^* \mathbb{E}[\mathbf{v'_2}]$ for any $\mathbb{E}$. It follows for any $\mathbb{E}$ that

$$
\begin{aligned}
&\mathbb{E}[\mathrm{upgrade}_{n+1;d}\ \mathbf{v}] \hookrightarrow^* \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathrm{upgrade}_{n;d}\ \mathbf{v.1}, \mathrm{upgrade}_{n;d}\ \mathbf{v.2} \rangle)] \hookrightarrow \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathrm{upgrade}_{n;d}\ \mathbf{v_1}, \mathrm{upgrade}_{n;d}\ \mathbf{v.2} \rangle)] \hookrightarrow^* \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v'_1}, \mathrm{upgrade}_{n;d}\ \mathbf{v.2} \rangle)] \hookrightarrow \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v'_1}, \mathrm{upgrade}_{n;d}\ \mathbf{v_2} \rangle)] \hookrightarrow^* \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v'_1}, \mathbf{v'_2} \rangle)]
\end{aligned}
$$

  and we have that $(\underline{W}, \mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v'_1}, \mathbf{v'_2} \rangle), \langle \mathsf{v}_1, \mathsf{v}_2 \rangle) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d+1;p}]\!]_{\square}$ by definition and by the fact that $\mathsf{lev}(\underline{W}) = 0$.

  If $\mathsf{lev}(\underline{W}) > 0$, then we have that $(\triangleright \underline{W}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$ and $(\triangleright \underline{W}, \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$.

  By induction, we have for any $\mathbb{E}$ that $\mathbb{E}[\mathrm{upgrade}_{n;d}\ \mathbf{v_1}] \hookrightarrow^* \mathbf{v'_1}$ and $\mathbb{E}[\mathrm{upgrade}_{n;d}\ \mathbf{v_2}] \hookrightarrow^* \mathbf{v'_2}$ for some $\mathbf{v'_1}, \mathbf{v'_2}$ with $(\triangleright \underline{W}, \mathbf{v'_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d;p}]\!]_{\square}$ and $(\triangleright \underline{W}, \mathbf{v'_2}, \mathsf{v}_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d;p}]\!]_{\square}$.

  We then also have for any $\mathbb{E}$ that

$$
\begin{aligned}
&\mathbb{E}[\mathrm{upgrade}_{n+1;d}\ \mathbf{v}] \hookrightarrow^* \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathrm{upgrade}_{n;d}\ \mathbf{v.1}, \mathrm{upgrade}_{n;d}\ \mathbf{v.2} \rangle)] \hookrightarrow \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathrm{upgrade}_{n;d}\ \mathbf{v_1}, \mathrm{upgrade}_{n;d}\ \mathbf{v.2} \rangle)] \hookrightarrow^* \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v'_1}, \mathrm{upgrade}_{n;d}\ \mathbf{v.2} \rangle)] \hookrightarrow \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v'_1}, \mathrm{upgrade}_{n;d}\ \mathbf{v_2} \rangle)] \hookrightarrow^* \\
&\quad \mathbb{E}[\mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v'_1}, \mathbf{v'_2} \rangle)]
\end{aligned}
$$

  and we have that $(\underline{W}, \mathbf{in}_{\times;\mathbf{n+d}}(\langle \mathbf{v'_1}, \mathbf{v'_2} \rangle), \langle \mathsf{v}_1, \mathsf{v}_2 \rangle) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+d+1;p}]\!]_{\square}$ by definition and by the facts that $(\triangleright \underline{W}, \mathbf{v'_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$ and $(\triangleright \underline{W}, \mathbf{v'_2}, \mathsf{v}_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_{\square}$.

- $\exists \mathbf{v'}.\, \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathbf{v'})$ and $(\underline{W}, \mathbf{v'}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n} \uplus \mathtt{EmulDV}_{n;p}]\!]_{\square}$. Similar to the previous case.

- $\exists \mathbf{v'}.\, \mathbf{v} = \mathbf{in}_{\rightarrow;\mathbf{n}}(\mathbf{v'})$ and $(\underline{W}, \mathbf{v'}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \rightarrow \mathtt{EmulDV}_{n;p}]\!]_{\square}$.

$$
\begin{aligned}
&\mathbb{E}[\mathrm{upgrade}_{n+1;d}\ \mathbf{v}] \hookrightarrow^* \\
&\quad \mathbb{E}[\mathbf{in}_{\rightarrow;\mathbf{n+d}}\ (\lambda \mathbf{z} : \mathrm{UVal}_{n+d}.\, \mathrm{upgrade}_{n;d}\ (\mathbf{v'}\ (\mathrm{downgrade}_{n;d}\ \mathbf{z})))]
\end{aligned}
$$

It remains to show that

$$(\underline{\mathsf{W}}, \lambda \mathbf{z} : \mathrm{UVal}_{\mathsf{n+d}}.\, \mathrm{upgrade}_{\mathsf{n;d}}\ (\mathbf{v}'\ (\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{z})), \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}} \to \mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}.$$

From $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \to \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$, it follows that $\mathbf{v}' = \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\, \mathbf{t}$ and $\mathsf{v} = \lambda \mathsf{x}.\, \mathsf{t}$ for some $\mathbf{t}$, t. Take $\underline{\mathsf{W}'} \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}'}, \mathbf{v}'', \mathsf{v}'') \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$, then we need to show that

$$(\underline{\mathsf{W}'}, \mathrm{upgrade}_{\mathsf{n;d}}\ (\mathbf{v}'\ (\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v}'')), \mathsf{t}[\mathsf{v}''/\mathsf{x}]) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}.$$

By induction, we get a $\mathbf{v}'''$ such that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{n;d}}\ \mathbf{v}''] \hookrightarrow^* \mathbb{E}[\mathbf{v}''']$ for any $\mathbb{E}$ and $(\underline{\mathsf{W}'}, \mathbf{v}''', \mathsf{v}'') \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$. We also have that $\mathbb{E}[\mathbf{v}'\ \mathbf{v}'''] \hookrightarrow \mathbb{E}[\mathbf{t}[\mathbf{v}'''/\mathbf{x}]]$. By Lemma 8, it suffices to prove that

$$(\underline{\mathsf{W}'}, \mathrm{upgrade}_{\mathsf{n;d}}\ (\mathbf{t}[\mathbf{v}'''/\mathbf{x}]), \mathsf{t}[\mathsf{v}''/\mathsf{x}]) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}.$$

Since we know that $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \to \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$, $\underline{\mathsf{W}'} \sqsupseteq \underline{\mathsf{W}}$ and $(\underline{\mathsf{W}'}, \mathbf{v}''', \mathsf{v}'') \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$, it follows that

$$(\underline{\mathsf{W}'}, \mathbf{t}[\mathbf{v}'''/\mathbf{x}], \mathsf{t}[\mathsf{v}''/\mathsf{x}]) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}.$$

By Lemma 19, it now suffices to show that for all $\underline{\mathsf{W}''} \sqsupseteq \underline{\mathsf{W}'}$, $(\underline{\mathsf{W}''}, \mathbf{v_4}, \mathsf{v}_4) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$, we have that $(\underline{\mathsf{W}''}, \mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v_4}, \mathsf{v}_4) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$. By induction, we get that $\mathbb{E}[\mathrm{upgrade}_{\mathsf{n;d}}\ \mathbf{v_4}] \hookrightarrow^* \mathbb{E}[\mathbf{v_5}]$ for any $\mathbb{E}$, for some $\mathbf{v_5}$ with $(\underline{\mathsf{W}''}, \mathbf{v_5}, \mathsf{v}_4) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$. By Lemma 8, it suffices to prove that $(\underline{\mathsf{W}''}, \mathbf{v_5}, \mathsf{v}_4) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{n+d;p}}]\!]_{\square}$, but this follows directly using Lemma 10.

$\square$

Theorem 9 (Upgrade and downgrade are semantics preserving)**.** *If ($n < m$ and $p = \mathtt{precise}$) or ($\square = \lesssim$ and $p = \mathtt{imprecise}$), and if $\mathbf{\Gamma} \vdash \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m+d;p}}$, then $\mathbf{\Gamma} \vdash \mathrm{downgrade}_{\mathsf{m;d}}\ \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$.*

If ($n < m$ and $p = \mathtt{precise}$) or ($\square = \lesssim$ and $p = \mathtt{imprecise}$), then if $\mathbf{\Gamma} \vdash \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$, then $\mathbf{\Gamma} \vdash \mathrm{upgrade}_{\mathsf{m;d}}\ \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m+d;p}}$.

Proof. Take $\mathbf{\Gamma} \vdash \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m+d;p}}$, $\underline{\mathsf{W}}$ with $\mathsf{lev}(\underline{\mathsf{W}}) \le n$ and $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$, then we need to prove that $(\underline{\mathsf{W}}, \mathrm{downgrade}_{\mathsf{m;d}}\ \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$.

From $\mathbf{\Gamma} \vdash \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m+d;p}}$, we have that $(\underline{\mathsf{W}}, \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+d;p}}]\!]_{\square}$. By Lemma 19, it then suffices to prove that for all $\underline{\mathsf{W}'} \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}'}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+d;p}}]\!]_{\square}$, we have that $(\underline{\mathsf{W}'}, \mathrm{downgrade}_{\mathsf{m;d}}\ \mathbf{v}, \mathsf{v}) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$.

We have that $\mathsf{lev}(\underline{\mathsf{W}'}) \le \mathsf{lev}(\underline{\mathsf{W}}) \le n$. By Lemma 37, there exists a $\mathbf{v}'$ such that $\mathbb{E}[\mathrm{downgrade}_{\mathsf{m;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ for any $\mathbb{E}$ and $(\underline{\mathsf{W}'}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$. By Lemma 8, it suffices to prove that $(\underline{\mathsf{W}'}, \mathbf{v}', \mathsf{v}) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$, but this follows directly from $(\underline{\mathsf{W}'}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ by Lemma 10.

Now take $\mathbf{\Gamma} \vdash \mathbf{t} \sqsupseteq_{\mathsf{n}} \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$, $\underline{\mathsf{W}}$ with $\mathsf{lev}(\underline{\mathsf{W}}) \leq n$ and $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}]\!]_{\square}$, then we need to prove that $(\underline{\mathsf{W}}, \mathrm{upgrade}_{\mathsf{m;d}}\ \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+d;p}}]\!]_{\square}$.

From $\mathbf{\Gamma} \vdash \mathbf{t} \sqsupseteq_{\mathsf{n}} \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$, we have that $(\underline{\mathsf{W}}, \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$. By Lemma 19, it then suffices to prove that for all $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$, we have that $(\underline{\mathsf{W}}', \mathrm{upgrade}_{\mathsf{m;d}}\ \mathbf{v}, \mathsf{v}) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+d;p}}]\!]_{\square}$.

We have that $\mathsf{lev}(\underline{\mathsf{W}}') \leq \mathsf{lev}(\underline{\mathsf{W}}) \leq n$. By Lemma 37, there exists a $\mathbf{v}'$ such that $\mathbb{E}[\mathrm{upgrade}_{\mathsf{m;d}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ for any $\mathbb{E}$ and $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+d;p}}]\!]_{\square}$. By Lemma 8, it suffices to prove that $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+d;p}}]\!]_{\square}$, but this follows directly from $(\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+d;p}}]\!]_{\square}$ by Lemma 10. ∎

6.4 Injecting λ^{τ} into UVal

$$
\begin{aligned}
\mathbf{extract}_{\tau;\mathsf{n}} &: \mathrm{UVal}_{\mathsf{n}} \to \tau \\
\mathbf{extract}_{\tau;0} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathbf{0}}.\, \mathrm{omega} \\
\mathbf{extract}_{\mathtt{Unit};\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+1}}.\, \mathtt{case}_{\mathtt{Unit};\mathtt{n}}\ \mathbf{x} \\
\mathbf{extract}_{\mathtt{Bool};\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+1}}.\, \mathtt{case}_{\mathtt{Bool};\mathtt{n}}\ \mathbf{x} \\
\mathbf{extract}_{\tau_1 \to \tau_2;\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+1}}.\, \lambda \mathbf{x} : \tau_1.\, \mathbf{extract}_{\tau_2;\mathsf{n}} \\
&\qquad\qquad (\mathtt{case}_{\to;\mathtt{n}}\ \mathbf{x}\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{x})) \\
\mathbf{extract}_{\tau_1 \times \tau_2;\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+1}}.\, \langle \mathbf{extract}_{\tau_1;\mathsf{n}}\ (\mathtt{case}_{\times;\mathtt{n}}\ \mathbf{x}).\mathbf{1}, \\
&\qquad\qquad \mathbf{extract}_{\tau_2;\mathsf{n}}\ (\mathtt{case}_{\times;\mathtt{n}}\ \mathbf{x}).\mathbf{2} \rangle \\
\mathbf{extract}_{\tau_1 \uplus \tau_2;\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n+1}}.\, \mathrm{case}\ \mathtt{case}_{\uplus;\mathtt{n}}\ \mathbf{x}\ \mathrm{of} \\
&\qquad \left|\ \begin{array}{l} \mathrm{inl}\ \mathbf{y} \to \mathrm{inl}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{y}) \\ \mathrm{inr}\ \mathbf{y} \to \mathrm{inr}\ (\mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathbf{y}) \end{array} \right.
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{inject}_{\tau;\mathsf{n}} &: \tau \to \mathrm{UVal}_{\mathsf{n}} \\
\mathbf{inject}_{\tau;0} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \tau.\, \mathrm{omega} \\
\mathbf{inject}_{\mathtt{Unit};\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathtt{Unit}.\, \mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ \mathbf{x} \\
\mathbf{inject}_{\mathtt{Bool};\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathtt{Bool}.\, \mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ \mathbf{x} \\
\mathbf{inject}_{\tau_1 \to \tau_2;\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \tau_1 \to \tau_2.\, \mathbf{in}_{\to;\mathbf{n}}\ (\lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}. \\
&\qquad \mathbf{inject}_{\tau_2;\mathsf{n}}\ (\mathbf{x}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{x}))) \\
\mathbf{inject}_{\tau_1 \times \tau_2;\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \tau_1 \times \tau_2.\, \mathbf{in}_{\times;\mathbf{n}} \langle \mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{x}.\mathbf{1}, \\
&\qquad\qquad \mathbf{inject}_{\tau_2;\mathsf{n}}\ \mathbf{x}.\mathbf{2} \rangle \\
\mathbf{inject}_{\tau_1 \uplus \tau_2;\mathsf{n+1}} &\stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \tau_1 \uplus \tau_2.\, \mathbf{in}_{\uplus;\mathbf{n}}\ (\mathrm{case}\ \mathbf{x}\ \mathrm{of} \\
&\qquad \left|\ \begin{array}{l} \mathrm{inl}\ \mathbf{y} \mapsto \mathrm{inr}\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{y}) \\ \mathrm{inr}\ \mathbf{y} \mapsto \mathrm{inr}\ (\mathbf{inject}_{\tau_2;\mathsf{n}}\ \mathbf{y}) \end{array} \right)
\end{aligned}
$$

Lemma 38 (Inject and extract are well-typed). *For all n, τ, $\mathbf{extract}_{\tau;\mathsf{n}} : \mathrm{UVal}_{\mathsf{n}} \to \tau$ and $\mathbf{inject}_{\tau;\mathsf{n}} : \tau \to \mathrm{UVal}_{\mathsf{n}}$.*

Proof. By definition. ∎

Lemma 39 (Diverging terms and non-values are related with no steps or for $\lesssim$)**.** *If* $\mathsf{lev}(\underline{\mathsf{W}}) = 0$ *or* $\square = \lesssim$*, if* $\mathbb{E}[\mathbf{t}]\Uparrow$ *for any* $\mathbb{E}$ *and* t *is not a value then* $(\mathbb{E}[\mathbf{t}], \mathbb{E}[\mathsf{t}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ *for any* $\mathbb{E}$*,* $\mathbb{E}$*.*

Proof. If $\mathsf{lev}(\underline{\mathsf{W}}) = 0$, then the result follows from Lemma 7 because $\mathbb{E}[\mathsf{t}]$ is not a value and neither is $\mathbb{E}[\mathbf{t}]$ since $\mathbb{E}[\mathbf{t}]\Uparrow$ for any $\mathbb{E}$.

If on the other hand $\square = \lesssim$, then we have that $(\mathbb{E}[\mathbf{t}], \mathbb{E}[\mathsf{t}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ by definition and by the fact that $\mathbb{E}[\mathbf{t}]\Uparrow$ for any $\mathbb{E}$. ☐

Lemma 40 (Inject/extract and protect/confine either relate at values or they are observationally equivalent)**.** *Assume that one of the following two conditions are fulfilled:*

- $n \geq \mathsf{lev}(\underline{\mathsf{W}})$ *and* $p = \mathtt{precise}$
- $\square = \lesssim$ *and* $p = \mathtt{imprecise}$

If $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau]\!]_\square$*, then one of the following holds:*

- *there exist* $\mathbf{v}'$ *and* v' *such that* $\mathbb{E}[\mathbf{inject}_{\tau;\mathsf{n}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ *and* $\mathbb{E}[\mathsf{protect}_\tau\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}']$ *for any* $\mathbb{E}$*,* $\mathbb{E}$ *and* $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_\square$*.*
- $(\mathbb{E}[\mathbf{inject}_{\tau;\mathsf{n}}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_\tau\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ *for any* $\mathbb{E}$*,* $\mathbb{E}$*.*

Also, if $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_\square$ *then one of the following must hold:*

- *there exist* $\mathbf{v}'$ *and* v' *such that* $\mathbb{E}[\mathbf{extract}_{\tau;\mathsf{n}}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ *and* $\mathbb{E}[\mathsf{confine}_\tau\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}']$ *for any* $\mathbb{E}$ *and* $\mathbb{E}$ *and we have that* $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\tau]\!]_\square$*.*
- $(\mathbb{E}[\mathbf{extract}_{\tau;\mathsf{n}}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_\tau\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ *for any* $\mathbb{E}$*,* $\mathbb{E}$*.*

Proof. We prove both results simultaneously, by induction on τ.

First, we consider the case that $n = 0$.

$$\mathbf{inject}_{\tau;0} = \lambda\mathbf{x} : \tau.\, \mathrm{omega}_{\mathrm{UVal}_\mathbf{0}}$$
$$\mathbf{extract}_{\tau;0} = \lambda\mathbf{x} : \mathrm{UVal}_\mathbf{0}.\, \mathrm{omega}_\tau$$

For $\mathbf{inject}_{\tau;0}$ and $\mathsf{protect}_\tau$, we have that $\mathsf{lev}(\underline{\mathsf{W}}) \leq n = 0$ or $\square = \lesssim$, that $\mathbb{E}[\mathbf{inject}_{\tau;0}\ \mathbf{v}]\Uparrow$ for any $\mathbb{E}$ and that $\mathsf{protect}_\tau\ \mathsf{v}$ is not a value, so by Lemma 39, it follows that $(\mathbb{E}[\mathbf{inject}_{\tau;0}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_\tau\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

For $\mathbf{extract}_{\tau;0}$ and $\mathsf{confine}_\tau$, almost exactly the same reasoning applies as for $\mathbf{inject}_{\tau;0}$ and $\mathsf{protect}_\tau$.

Now consider the case for $n + 1$. We do a case analysis on τ.

- $\tau = \mathcal{B}$: We have that

$$\mathsf{protect}_\mathcal{B} = \lambda\mathsf{x}.\, \mathsf{x}$$
$$\mathsf{confine}_{\mathtt{Unit}} \stackrel{\mathsf{def}}{=} \lambda\mathsf{y}.\, \mathsf{y}; \mathtt{unit}$$
$$\mathsf{confine}_{\mathtt{Bool}} \stackrel{\mathsf{def}}{=} \lambda\mathsf{y}.\, \mathrm{if}\ \mathsf{y}\ \mathrm{then}\ \mathtt{true}\ \mathrm{else}\ \mathtt{false}$$
$$\mathbf{extract}_{\mathcal{B};\mathsf{n}+1} = \lambda\mathbf{x} : \mathrm{UVal}_{\mathsf{n}+1}.\, \mathtt{case}_{\mathcal{B};\mathsf{n}}\ \mathbf{x}$$
$$\mathbf{inject}_{\mathcal{B};\mathsf{n}+1} = \lambda\mathbf{x} : \mathbf{b}.\, \mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{x}$$

For $\mathsf{protect}_{\mathcal{B}}$, we directly have that $\mathbb{E}[\mathsf{protect}_{\mathcal{B}}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\mathsf{v}]$ for any $\mathbb{E}$. We also have that $\mathbb{E}[\mathbf{inject}_{\mathcal{B};\mathsf{n}+1}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{v}]$ for any $\mathbb{E}$, so we can take $\mathbf{v}' = \mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{v}$, $\mathsf{v}' = \mathsf{v}$. It remains to prove that $(\underline{\mathsf{W}}, \mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{v}, \mathsf{v}) \in \mathtt{EmulDV}_{\mathsf{n}+1;\mathsf{p}}$. This follows directly from the definition of $\mathtt{EmulDV}_{\mathsf{n}+1;\mathsf{p}}$, since we have that $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$.

For $\mathsf{confine}_{\mathcal{B}}$, we get from $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathtt{EmulDV}_{\mathsf{n}+1;\mathsf{p}}$ that one of five cases holds:

$$\begin{cases} \mathbf{v} = \mathbf{in}_{\mathrm{unk};\mathbf{n}} \wedge p = \mathtt{imprecise} \\ \exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\mathcal{B};\mathbf{n}}(v') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square} \\ \exists \mathbf{v}', m'.\, \mathbf{v} = \mathbf{in}_{\times;\mathbf{n}}(\mathbf{v}') \wedge (m = m'+1 \vee m = m' = 0) \wedge \\ \qquad (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \times \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square} \\ \exists \mathbf{v}', m'.\, \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathbf{v}') \wedge (m = m'+1 \vee m = m' = 0) \wedge \\ \qquad (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \uplus \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square} \\ \exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\to;\mathbf{n}}(\mathbf{v}') \wedge \\ \quad \forall m' < m.\, (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \to \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square} \end{cases}$$

In the first case, we know that $\square = \lesssim$ from the assumptions, $\mathbb{E}[\mathbf{extract}_{\tau;\mathsf{n}+1}\ \mathbf{v}]\Uparrow$ for any $\mathbb{E}$ and $\mathsf{confine}_{\tau}\ \mathsf{v}$ is not a value, so that by definition of $\mathsf{O}(\underline{\mathsf{W}})_{\lesssim}$, we have that $(\mathbb{E}[\mathbf{extract}_{\tau;\mathsf{n}+1}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_{\square}$ for any $\mathbb{E}$, $\mathbb{E}$.

Next, we distinguish the second case and the three others. In fact, within the second case, (where $\mathbf{v} = \mathbf{in}_{\mathcal{B}';\mathbf{n}}(v')$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}']\!]_{\square}$), there is the case that $\mathcal{B} = \mathcal{B}'$ and $\mathcal{B} \neq \mathcal{B}'$. We treat the former specially and deal with the latter together with the three other top-level cases.

So, first, assume that $\mathbf{v} = \mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{v}'$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$. This implies that $\mathbf{v}' = \mathsf{v} = \mathtt{unit}$ if $\mathcal{B} = \mathtt{Unit}$ and $\mathbf{v}' = \mathsf{v} = v$ for some $v \in \{\mathtt{true}, \mathtt{false}\}$ if $\mathcal{B} = \mathtt{Bool}$.

It follows for any $\mathbb{E}$, $\mathbb{E}$ that

$$\mathbb{E}[\mathsf{confine}_{\mathcal{B}}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\mathsf{v}]$$

and

$$\begin{aligned} &\mathbb{E}[\mathbf{extract}_{\mathcal{B};\mathsf{n}+1}\ \mathbf{v}] = \mathbb{E}[\mathtt{case}_{\mathcal{B},\mathsf{n}}\ \mathbf{v}] = \\ &\quad \mathbb{E}[(\lambda uv : \mathrm{UVal}_{\mathsf{n}+1}.\, \mathrm{case}\ uv\ \mathrm{of}\ \{\mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{\mathcal{B}}\})\ \mathbf{v}] \hookrightarrow \\ &\qquad\qquad \mathbb{E}[\mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{\mathcal{B}}\}] = \\ &\qquad\qquad \mathbb{E}[\mathrm{case}\ (\mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{v}')\ \mathrm{of}\ \{\mathbf{in}_{\mathcal{B};\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{\mathcal{B}}\}] \hookrightarrow \mathbb{E}[\mathbf{v}'] \end{aligned}$$

Since we already know that $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$, this case is done.

Secondly, we assume that $\mathcal{B} \neq \mathcal{B}'$ or $\mathbf{v} = \mathbf{in}_{\times;\mathbf{n}}(\mathbf{v}')$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \times \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$ or $\mathbf{v} = \mathbf{in}_{\to;\mathbf{n}}(\mathbf{v}')$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \to \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$ or $\mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathbf{v}')$ and $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \uplus \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$. In the first case, we have that $\mathcal{B} = \mathtt{Bool}$,$\mathcal{B}' = \mathtt{Unit}$ and $\mathsf{v} = \mathtt{unit}$ or $\mathcal{B} = \mathtt{Unit}$,

$\mathcal{B}' = \mathtt{Bool}$ and $\mathsf{v} \in \{\mathtt{true}, \mathtt{false}\}$. In the second case, we have that $\mathsf{v} = \langle \mathsf{v}_1, \mathsf{v}_2 \rangle$ for some $\mathsf{v}_1, \mathsf{v}_2$, in the third case $\mathsf{v} = \lambda \mathsf{x}.\ \mathsf{t}$ for some t and in the fourth case $\mathsf{v} = \mathrm{inl}\ \mathsf{v}_1$ or $\mathsf{v} = \mathrm{inl}\ \mathsf{v}_2$ for some v_1 or v_2.

From this, it follows for any $\mathbb{E}$ and $\mathbb{E}$ that

$$\mathbb{E}[\mathsf{confine}_{\mathcal{B}}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\mathsf{wrong}] \hookrightarrow \mathsf{wrong}$$

and

$$\mathbb{E}[\mathbf{extract}_{\mathcal{B};\mathsf{n}+1}\ \mathbf{v}] = \mathbb{E}[\mathtt{case}_{\mathcal{B};\mathsf{n}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathrm{omega}_{\mathcal{B}}]$$

We know that $\mathbb{E}[\mathrm{omega}_{\mathcal{B}}]\Uparrow$ (by Lemma 34) for any evaluation contexts $\mathbb{E}$, so that we get by Lemma 6 that $(\mathbb{E}[\mathbf{extract}_{\mathcal{B};\mathsf{n}+1}\ \mathbf{v}], \mathbb{E}[\mathbf{extract}_{\mathcal{B};\mathsf{n}+1}\ \mathbf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})$ for any $\mathbb{E}$, $\mathbb{E}$.

- $\tau = \tau_1 \to \tau_2$: We have that

$$\mathsf{protect}_{\tau_1 \to \tau_2} = \lambda \mathsf{y}.\ \lambda \mathsf{x}.\mathsf{protect}_{\tau_2}\ (\mathsf{y}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}))$$
$$\mathsf{confine}_{\tau_1 \to \tau_2} = \lambda \mathsf{y}.\ \lambda \mathsf{x}.\ \mathsf{confine}_{\tau_2}\ (\mathsf{y}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x}))$$
$$\mathbf{extract}_{\tau_1 \to \tau_2;\mathsf{n}+1} = \lambda uv : \mathrm{UVal}_{\mathsf{n}+1}.\ \lambda \mathbf{x} : \tau_1.\ \mathbf{extract}_{\tau_2;\mathsf{n}}\ (\mathtt{case}_{\to;\mathsf{n}}\ uv\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{x}))$$
$$\mathbf{inject}_{\tau_1 \to \tau_2;\mathsf{n}+1} = \lambda \mathbf{v} : \tau_1 \to \tau_2.\ \mathbf{in}_{\to;\mathbf{n}}\ (\lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\ \mathbf{inject}_{\tau_2;\mathsf{n}}\ (\mathbf{v}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{x}))).$$

  - First, we consider $\mathsf{protect}_{\tau_1 \to \tau_2}$ and $\mathbf{inject}_{\tau_1 \to \tau_2;\mathsf{n}+1}$. We have for any $\mathbb{E}$ that

$$\mathbb{E}[\mathsf{protect}_{\tau_1 \to \tau_2}\ \mathsf{v}] = \mathbb{E}[(\lambda \mathsf{y}.\ \lambda \mathsf{x}.\mathsf{protect}_{\tau_2}\ (\mathsf{y}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x})))\ \mathsf{v}] \hookrightarrow \mathbb{E}[\lambda \mathsf{x}.\ \mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}))]$$

  and for any $\mathbb{E}$

$$\begin{aligned}&\mathbb{E}[\mathbf{inject}_{\tau_1 \to \tau_2;\mathsf{n}+1}\ \mathbf{v}] = \\ &\mathbb{E}[(\lambda \mathbf{v} : \tau_1 \to \tau_2.\ \mathbf{in}_{\to;\mathbf{n}}\ (\lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\ \mathbf{inject}_{\tau_2;\mathsf{n}}\ (\mathbf{v}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{x}))))\ \mathbf{v}] \hookrightarrow \\ &\qquad \mathbb{E}[\mathbf{in}_{\to;\mathbf{n}}\ (\lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\ \mathbf{inject}_{\tau_2;\mathsf{n}}\ (\mathbf{v}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{x})))].\end{aligned}$$

  We take

$$\mathsf{v}' = \lambda \mathsf{x}.\ \mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}))$$

  and

$$\mathbf{v}' = \mathbf{in}_{\to;\mathbf{n}}\ (\lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\ \mathbf{inject}_{\tau_2;\mathsf{n}}\ (\mathbf{v}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{x})))$$

  and it remains to prove that $(\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n}+1;\mathsf{p}}]\!]_{\square}$. Define $\mathbf{v}'' = \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\ \mathbf{inject}_{\tau_2;\mathsf{n}}\ (\mathbf{v}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{x}))$. By definition of $\mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n}+1;\mathsf{n}}]\!]_{\square}$, it suffices to show that $(\underline{\mathsf{W}}, \mathbf{v}'', \mathsf{v}') \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \to \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$. We need to prove that $\mathbf{v}''$ is well typed ($\mathbf{oftype}()$ condition of the logical relations), which follows from Lemma 38 and rule λ^{τ}-Type-fun.

Now take $\underline{W'} \sqsupset \underline{W}$ and $(\underline{W'}, \mathbf{v'''}, \mathsf{v'''}) \in \mathcal{V}[\![\texttt{EmulDV}_{n;p}]\!]_\square$. It suffices to show that

$$(\underline{W'}, \mathbf{inject}_{\tau_2,n}\ (\mathbf{v}\ (\mathbf{extract}_{\tau_1;n}\ \mathbf{v'''})),\\ \mathsf{protect}_{\tau_2}\ (\mathsf{v}\ (\mathsf{confine}_{\tau_1}\ \mathsf{v'''}))) \in \mathcal{E}[\![\texttt{EmulDV}_{n;p}]\!]_\square.$$

By induction, we have that one of the following cases holds:

- there exist $\mathbf{v''''}$ and $\mathsf{v''''}$ such that $\mathbb{E}[\mathbf{extract}_{\tau_1;n}\ \mathbf{v'''}] \hookrightarrow^* \mathbb{E}[\mathbf{v''''}]$ and $\mathbb{E}[\mathsf{confine}_{\tau_1}\ \mathsf{v'''}] \hookrightarrow^* \mathbb{E}[\mathsf{v''''}]$ for any $\mathbb{E}$, $\mathbb{E}$ and $(\underline{W'}, \mathbf{v''''}, \mathsf{v''''}) \in \mathcal{V}[\![\tau_1]\!]_\square$
- $(\mathbb{E}[\mathbf{extract}_{\tau;n}\ \mathbf{v'''}], \mathbb{E}[\mathsf{confine}_{\tau}\ \mathsf{v'''}]) \in \mathsf{O}(\underline{W})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the latter case, the result follows easily from the definition of $\mathcal{E}[\![\cdots]\!]_\square$. In the former case, by Lemma 4 it suffices to prove that

$$(\underline{W'}, \mathbf{inject}_{\tau_2;n}\ (\mathbf{v}\ \mathbf{v''''}), \mathsf{protect}_{\tau_2}\ (\mathsf{v}\ \mathsf{v''''})) \in \mathcal{E}[\![\texttt{EmulDV}_{n;p}]\!]_\square.$$

By Lemma 20, we have that $(\underline{W'}, \mathbf{v}\ \mathbf{v''''}, \mathsf{v}\ \mathsf{v''''}) \in \mathcal{E}[\![\tau_2]\!]_\square$ since $(\underline{W'}, \mathbf{v''''}, \mathsf{v''''}) \in \mathcal{V}[\![\tau_1]\!]_\square$ and we get $(\underline{W'}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]_\square$ from $(\underline{W}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \to \tau_2]\!]_\square$ by Lemma 13.

By Lemma 19, it then suffices to prove that for all $\underline{W''} \sqsupseteq \underline{W'}$, $(\underline{W''}, \mathbf{v_5}, \mathsf{v_5}) \in \mathcal{V}[\![\tau_2]\!]_\square$, we have that $(\underline{W''}, \mathbf{inject}_{\tau_2;n}\ \mathbf{v_5}, \mathsf{protect}_{\tau_2}\ \mathsf{v_5}) \in \mathcal{E}[\![\texttt{EmulDV}_{n;p}]\!]_\square$.
Again by induction, we know that one of the following cases holds:

- there exist $\mathbf{v_6}$ and $\mathsf{v_6}$ such that $\mathbb{E}[\mathbf{inject}_{\tau_2;n}\ \mathbf{v_5}] \hookrightarrow^* \mathbb{E}[\mathbf{v_6}]$ and $\mathbb{E}[\mathsf{protect}_{\tau_2}\ \mathsf{v_5}] \hookrightarrow^* \mathbb{E}[\mathsf{v_6}]$ and $(\underline{W''}, \mathbf{v_6}, \mathsf{v_6}) \in \mathcal{V}[\![\texttt{EmulDV}_{n;p}]\!]_\square$. The result then follows by Lemmas 8 and 10.
- $(\mathbb{E}[\mathbf{inject}_{\tau_2;n}\ \mathbf{v_5}], \mathbb{E}[\mathsf{protect}_{\tau_2}\ \mathsf{v_5}]) \in \mathsf{O}(\underline{W''})_\square$ for any $\mathbb{E}$, $\mathbb{E}$. The result follows by unfolding the definition of $\mathcal{E}[\![\texttt{EmulDV}_{n;p}]\!]_\square$.

- Next, we consider $\mathsf{confine}_{\tau_1 \to \tau_2}$ and $\mathbf{extract}_{\tau_1 \to \tau_2;n+1}$. We have that

$$\mathbb{E}[\mathsf{confine}_{\tau_1 \to \tau_2}\ \mathsf{v}] =\\ \mathbb{E}[(\lambda \mathsf{y}.\ \lambda \mathsf{x}.\ \mathsf{confine}_{\tau_2}\ (\mathsf{y}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x})))\ \mathsf{v}] \hookrightarrow\\ \mathbb{E}[\lambda \mathsf{x}.\ \mathsf{confine}_{\tau_2}\ (\mathsf{v}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x}))]$$

for any $\mathbb{E}$ and

$$\mathbb{E}[\mathbf{extract}_{\tau_1 \to \tau_2;n+1}\ \mathbf{v}] =\\ \mathbb{E}[(\lambda uv : \mathrm{UVal}_{n+1}.\ \lambda \mathbf{x} : \tau_1.\ \mathbf{extract}_{\tau_2;n}\ (\mathsf{case}_{\to;n}\ uv\ (\mathbf{inject}_{\tau_1;n}\ \mathbf{x})))\ \mathbf{v}] \hookrightarrow\\ \mathbb{E}[\lambda \mathbf{x} : \tau_1.\ \mathbf{extract}_{\tau_2;n}\ (\mathsf{case}_{\to;n}\ \mathbf{v}\ (\mathbf{inject}_{\tau_1;n}\ \mathbf{x}))]$$

for any $\mathbb{E}$.
We take

$$\mathsf{v'} = \lambda \mathsf{x}.\ \mathsf{confine}_{\tau_2}\ (\mathsf{v}\ (\mathsf{protect}_{\tau_1}\ \mathsf{x}))$$

and

$$\mathbf{v'} = \lambda \mathbf{x} : \tau_{\mathbf{1}}.\, \mathbf{extract}_{\tau_2;\mathsf{n}}\ (\mathtt{case}_{\to;\mathsf{n}}\ \mathbf{v}\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{x}))$$

and it suffices to prove that $(\underline{\mathsf{W}}, \mathbf{v'}, \mathsf{v'}) \in \mathcal{V}[\![\tau_{\mathbf{1}} \to \tau_{\mathbf{2}}]\!]_{\square}$.
We need to prove that $\mathbf{v'}$ is well typed ($\mathbf{oftype}()$ condition of the logical relations) that follows from Lemma 38 and rule λ^{τ}-Type-fun.
Now take $\underline{\mathsf{W}}' \sqsupset \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v_2}, \mathsf{v_2}) \in \mathcal{V}[\![\tau_{\mathbf{1}}]\!]_{\square}$, then we need to prove that

$$\begin{aligned}(\underline{\mathsf{W}}', &\mathbf{extract}_{\tau_2;\mathsf{n}}\ (\mathtt{case}_{\to;\mathsf{n}}\ \mathbf{v}\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_2})),\\ &\qquad\qquad \mathsf{confine}_{\tau_2}\ (\mathsf{v}\ (\mathsf{protect}_{\tau_1}\ \mathsf{v_2}))) \in \mathcal{E}[\![\tau_{\mathbf{2}}]\!]_{\square}.\end{aligned}$$

We have that

$$\mathtt{case}_{\to;\mathsf{n}} = \lambda uv : \mathrm{UVal}_{\mathsf{n}+1}.\, \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{case}\ uv\ \mathrm{of}\ \{\mathbf{in}_{\to;\mathbf{n}}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{x};\ \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_{\mathsf{n}}}\},$$

so that

$$\begin{aligned}&\mathbf{extract}_{\tau_2;\mathsf{n}}\ (\mathtt{case}_{\to;\mathsf{n}}\ \mathbf{v}\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_2})) =\\ &\mathbf{extract}_{\tau_2;\mathsf{n}}\ ((\lambda uv : \mathrm{UVal}_{\mathsf{n}+1}.\, \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{case}\ uv\ \mathrm{of}\ \{\mathbf{in}_{\to;\mathbf{n}}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{x};\\ &\qquad\qquad \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_{\mathsf{n}}}\})\ \mathbf{v}\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_2})) \hookrightarrow\\ &\qquad \mathbf{extract}_{\tau_2;\mathsf{n}}\ ((\lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\to;\mathbf{n}}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{x};\\ &\qquad\qquad \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_{\mathsf{n}}}\})\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_2}))\end{aligned}$$

We call

$$\mathbf{v'} \stackrel{\mathsf{def}}{=} \lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\to;\mathbf{n}}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{x};\ \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_{\mathsf{n}}}\}$$

and by Lemma 4 and some definition unfolding, it suffices to prove that

$$\begin{aligned}(\underline{\mathsf{W}}', &\mathbf{extract}_{\tau_2;\mathsf{n}}\ (\mathbf{v'}\ (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_2})),\\ &\qquad\qquad \mathsf{confine}_{\tau_2}\ (\mathsf{v}\ (\mathsf{protect}_{\tau_1}\ \mathsf{v_2}))) \in \mathcal{E}[\![\tau_{\mathbf{2}}]\!]_{\square}.\end{aligned}$$

By induction, we have that one of the following holds:

- there exist $\mathbf{v_3}$,$\mathsf{v_3}$ such that $\mathbb{E}[\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_2}] \hookrightarrow^* \mathbb{E}[\mathbf{v_3}]$ and $\mathbb{E}[\mathsf{protect}_{\tau_1}\ \mathsf{v_2}] \hookrightarrow^* \mathbb{E}[\mathsf{v_3}]$ for any $\mathbb{E}$, $\mathbb{E}$ and $(\underline{\mathsf{W}}', \mathbf{v_3}, \mathsf{v_3}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$.
- $(\mathbb{E}[\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_2}], \mathbb{E}[\mathsf{protect}_{\tau_1}\ \mathsf{v_2}]) \in \mathsf{O}(\underline{\mathsf{W}}')_{\square}$ for any $\mathbb{E}$, $\mathbb{E}$.

In the latter case, the result follows by unfolding the definition of $\mathcal{E}[\![\tau_{\mathbf{2}}]\!]_{\square}$.
In the former case, by Lemma 8 it suffices to prove that

$$(\underline{\mathsf{W}}', \mathbf{extract}_{\tau_2;\mathsf{n}}\ (\mathbf{v'}\ \mathbf{v_3}), \mathsf{confine}_{\tau_2}\ (\mathsf{v}\ \mathsf{v_3})) \in \mathcal{E}[\![\tau_{\mathbf{2}}]\!]_{\square}.$$

We have that

$$\mathbf{extract}_{\tau_2;n}\ (\mathbf{v}'\ \mathbf{v_3}) =$$
$$\mathbf{extract}_{\tau_2;n}\ ((\lambda\mathbf{x} : \mathrm{UVal}_n.\, \mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\rightarrow;n}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{x}; \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_n}\})\ \mathbf{v_3}) \hookrightarrow$$
$$\mathbf{extract}_{\tau_2;n}\ (\mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\rightarrow;n}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{v_3}; \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_n}\}) \hookrightarrow$$

and again by Lemma 8, it suffices to prove that

$$(\underline{W}', \mathbf{extract}_{\tau_2;n}\ (\mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\rightarrow;n}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{v_3}; \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_n}\}), \mathsf{confine}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}_3)) \in \mathcal{E}[\![\tau_2]\!]_\square.$$

Now, from $(\underline{W}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+1;p}]\!]_\square$, we get that one of the following must hold:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};n} \wedge p = \mathtt{imprecise}$
- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\mathcal{B};n}(\mathbf{v}')$ and $(\underline{W}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_\square$
- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\times;n}(\mathbf{v}') \wedge (\underline{W}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \times \mathtt{EmulDV}_{n;p}]\!]_\square$
- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\uplus;n}(\mathbf{v}') \wedge (\underline{W}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \uplus \mathtt{EmulDV}_{n;p}]\!]_\square$
- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\rightarrow;n}(\mathbf{v}') \wedge (\underline{W}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p} \rightarrow \mathtt{EmulDV}_{n;p}]\!]_\square$

In the first case, we have that $\square = \lesssim$ and we know that

$$\mathbb{E}[\mathbf{extract}_{\tau_2;n}\ (\mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\rightarrow;n}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{v_3}; \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_n}\})] \hookrightarrow \mathbb{E}[\mathbf{extract}_{\tau_2;n}\ \mathrm{omega}_{\mathrm{UVal}_n}]$$

which diverges for any $\mathbb{E}$. It follows by definition of $\mathsf{O}(\underline{W})_\lesssim$ and $\mathcal{E}[\![\tau_2]\!]_\square$ that

$$(\underline{W}', \mathbf{extract}_{\tau_2;n}\ (\mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\rightarrow;n}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{v_3}; \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_n}\}), \mathsf{confine}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}_3)) \in \mathcal{E}[\![\tau_2]\!]_\square.$$

In the second, third and fourth case, we have that

$$\mathbb{E}[\mathbf{extract}_{\tau_2;n}\ (\mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\rightarrow;n}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{v_3}; \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_n}\})] \hookrightarrow \mathbb{E}[\mathbf{extract}_{\tau_2;n}\ \mathrm{omega}_{\mathrm{UVal}_n}]$$

for any $\mathbb{E}$ and $\mathbb{E}[\mathsf{confine}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}_3)] \hookrightarrow \mathbb{E}[\mathsf{confine}_{\tau_2}\ \mathsf{wrong}]$ for any $\mathbb{E}$. This means that $\mathbb{E}[\mathbf{extract}_{\tau_2;n}\ \mathrm{omega}_{\mathrm{UVal}_n}]\Uparrow$ for any $\mathbb{E}$ and $\mathbb{E}[\mathsf{confine}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}_3)] \hookrightarrow^* \mathsf{wrong}$ for any $\mathbb{E}$. By Lemma 6, we have that $(\mathbb{E}[\mathbf{extract}_{\tau_2;n}\ \mathrm{omega}_{\mathrm{UVal}_n}], \mathbb{E}[\mathsf{confine}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}_3)]) \in \mathsf{O}(\underline{W}')$ for any $\mathbb{E}$, $\mathbb{E}$. The result follows from the above evaluations, Lemma 4 and the definition of $\mathcal{E}[\![\tau_2]\!]_\square$.

In the last case, we have that

$$\mathbf{extract}_{\tau_2;n}\ (\mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\rightarrow;n}\ \mathbf{y} \mapsto \mathbf{y}\ \mathbf{v_3}; \_ \mapsto \mathrm{omega}_{\mathrm{UVal}_n}\}) \hookrightarrow \mathbf{extract}_{\tau_2;n}\ (\mathbf{v}''\ \mathbf{v_3})$$

with $(\underline{W}, \mathbf{v}'', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV_{n;p}} \to \mathtt{EmulDV_{n;p}}]\!]_\square$. Again by Lemma 8, it suffices to prove that

$$(\underline{W}', \mathbf{extract}_{\tau_2;n}\ (\mathbf{v}''\ \mathbf{v_3}), \mathsf{confine}_{\tau_2}\ (\mathsf{v}\ \mathsf{v}_3)) \in \mathcal{E}[\![\tau_2]\!]_\square.$$

By the facts that $(\underline{W}, \mathbf{v}'', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV_{n;p}} \to \mathtt{EmulDV_{n;p}}]\!]_\square$, $(\underline{W}', \mathbf{v_3}, \mathsf{v}_3) \in \mathcal{V}[\![\mathtt{EmulDV_{n;p}}]\!]_\square$, by Lemmas 13 and 20, we have that $(\underline{W}', \mathbf{v}''\ \mathbf{v_3}, \mathsf{v}\ \mathsf{v}_3) \in \mathcal{E}[\![\mathtt{EmulDV_{n;p}}]\!]_\square$. By Lemma 19, it suffices to prove for $\underline{W}'' \sqsupseteq \underline{W}'$, $(\underline{W}'', \mathbf{v_4}, \mathsf{v}_4) \in \mathcal{V}[\![\mathtt{EmulDV_{n;p}}]\!]_\square$ that

$$(\underline{W}'', \mathbf{extract}_{\tau_2;n}\ \mathbf{v_4}, \mathsf{confine}_{\tau_2}\ \mathsf{v}_4) \in \mathcal{E}[\![\tau_2]\!]_\square.$$

By induction, we have that one of the following must hold:

- there exist $\mathbf{v_5}$ and v_5 such that $\mathbb{E}[\mathbf{extract}_{\tau_2;n}\ \mathbf{v_4}] \hookrightarrow^* \mathbb{E}[\mathbf{v_5}]$ and $\mathbb{E}[\mathsf{confine}_{\tau_2}\ \mathsf{v}_4] \hookrightarrow^* \mathbb{E}[\mathsf{v}_5]$ for any $\mathbb{E}$ and $\mathbb{E}$ and $(\underline{W}, \mathbf{v_5}, \mathsf{v}_5) \in \mathcal{V}[\![\tau_2]\!]_\square$
- $(\mathbb{E}[\mathbf{extract}_{\tau_2;n}\ \mathbf{v_4}], \mathbb{E}[\mathsf{confine}_{\tau_2}\ \mathsf{v}_4]) \in \mathsf{O}(\underline{W})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the latter case, the result follows directly by definition of $\mathcal{E}[\![\tau_2]\!]_\square$. In the former case, the result follows by Lemma 8 and Lemma 10.

- $\tau = \tau_1 \times \tau_2$: We have that

$$\begin{aligned}
\mathbf{inject}_{\tau_1\times\tau_2;n+1} &= \lambda\mathbf{v} : \tau_1 \times \tau_2.\ \mathbf{in}_{\times;n}\langle \mathbf{inject}_{\tau_1;n}\ \mathbf{v.1}, \mathbf{inject}_{\tau_2;n}\ \mathbf{v.2}\rangle \\
\mathbf{extract}_{\tau_1\times\tau_2;n+1} &= \lambda uv : \mathrm{UVal}_{n+1}.\ \langle \mathbf{extract}_{\tau_1;n}\ \mathtt{case}_{\times;n}\ uv.\mathbf{1}, \mathbf{extract}_{\tau_2;n}\ \mathtt{case}_{\times;n}\ uv.\mathbf{2}\rangle \\
\mathsf{protect}_{\tau_1\times\tau_2} &= \lambda \mathsf{y}.\ \langle \mathsf{protect}_{\tau_1}\ \mathsf{y.1}, \mathsf{protect}_{\tau_2}\ \mathsf{y.2}\rangle \\
\mathsf{confine}_{\tau_1\times\tau_2} &\stackrel{\mathsf{def}}{=} \lambda \mathsf{y}.\ \langle \mathsf{confine}_{\tau_1}\ \mathsf{y.1}, \mathsf{confine}_{\tau_2}\ \mathsf{y.2}\rangle
\end{aligned}$$

If $(\underline{W}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \times \tau_2]\!]_\square$, then we have that $\mathbf{v} = \langle \mathbf{v_1}, \mathbf{v_2}\rangle$ and $\mathsf{v} = \langle \mathsf{v}_1, \mathsf{v}_2\rangle$ for some $\mathbf{v_1}, \mathbf{v_2}$, $\mathsf{v}_1, \mathsf{v}_2$ with $(\underline{W}, \mathbf{v_1}, \mathsf{v}_1) \in \triangleright\mathcal{V}[\![\tau_1]\!]_\square$ and $(\underline{W}, \mathbf{v_2}, \mathsf{v}_2) \in \triangleright\mathcal{V}[\![\tau_2]\!]_\square$.

If $\mathsf{lev}(\underline{W}) = 0$, then we know by Lemma 7 that $(\mathbb{E}[\mathbf{inject}_{\tau_1\times\tau_2;n+1}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_{\tau_1\times\tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{W})_\square$ for any $\mathbb{E}$, $\mathbb{E}$, since $\mathbf{inject}_{\tau_1\times\tau_2;n+1}\ \mathbf{v}$ and $\mathsf{protect}_{\tau_1\times\tau_2}\ \mathsf{v}$ are not values.

If $\mathsf{lev}(\underline{W}) > 0$, then we know that $(\triangleright\underline{W}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\tau_1]\!]_\square$ and $(\triangleright\underline{W}, \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\tau_2]\!]_\square$. We have for any $\mathbb{E}$ that

$$\begin{aligned}
&\mathbb{E}[\mathbf{inject}_{\tau_1\times\tau_2;n+1}\ \mathbf{v}] \hookrightarrow \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;n}\langle \mathbf{inject}_{\tau_1;n}\ \mathbf{v.1}, \mathbf{inject}_{\tau_2;n}\ \mathbf{v.2}\rangle] \hookrightarrow \\
&\qquad\qquad \mathbb{E}[\mathbf{in}_{\times;n}\langle \mathbf{inject}_{\tau_1;n}\ \mathbf{v_1}, \mathbf{inject}_{\tau_2;n}\ \mathbf{v.2}\rangle]
\end{aligned}$$

and for any $\mathbb{E}$ that

$$\begin{aligned}
&\mathbb{E}[\mathsf{protect}_{\tau_1\times\tau_2}\ \mathsf{v}] \hookrightarrow \\
&\qquad \mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v.1}, \mathsf{protect}_{\tau_2}\ \mathsf{v.2}\rangle] \hookrightarrow \\
&\qquad\qquad \mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}_1, \mathsf{protect}_{\tau_2}\ \mathsf{v.2}\rangle].
\end{aligned}$$

By the induction hypothesis for τ_1, we have that one of the following must hold:

- there are $\mathbf{v}'_1$ and v'_1 such that $\mathbb{E}[\mathbf{inject}_{\tau_1;n}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v}'_1]$ and $\mathbb{E}[\mathsf{protect}_{\tau_1}\ \mathsf{v}_1] \hookrightarrow^* \mathbb{E}[\mathsf{v}'_1]$ for any $\mathbb{E}$ and $\mathbb{E}$ and that $(\triangleright \underline{\mathsf{W}}, \mathbf{v}'_1, \mathsf{v}'_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_\square$.
- $(\mathbb{E}[\mathbf{inject}_{\tau_1;n}\ \mathbf{v_1}], \mathbb{E}[\mathsf{protect}_{\tau_1}\ \mathsf{v}_1]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}})_\square$ and for any $\mathbb{E}$, $\mathbb{E}$.

In the latter case, we have by the above evaluation and by Lemma 4 that $(\mathbb{E}[\mathbf{inject}_{\tau_1\times\tau_2;n+1}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_{\tau_1\times\tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the former case, we can continue the evaluations for any $\mathbb{E}$ and for any $\mathbb{E}$ as follows:

$$\begin{aligned}
&\mathbb{E}[\mathbf{in}_{\times;n}\langle \mathbf{inject}_{\tau_1;n}\ \mathbf{v_1}, \mathbf{inject}_{\tau_2;n}\ \mathbf{v.2}\rangle] \hookrightarrow^* \\
&\qquad \mathbb{E}[\mathbf{in}_{\times;n}\langle \mathbf{v}'_1, \mathbf{inject}_{\tau_2;n}\ \mathbf{v.2}\rangle] \hookrightarrow \\
&\qquad\qquad \mathbb{E}[\mathbf{in}_{\times;n}\langle \mathbf{v}'_1, \mathbf{inject}_{\tau_2;n}\ \mathbf{v_2}\rangle]
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}[\langle \mathsf{protect}_{\tau_1}\ \mathsf{v}_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow^* \\
&\qquad \mathbb{E}[\langle \mathsf{v}'_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow \mathbb{E}[\langle \mathsf{v}'_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}_2\rangle]
\end{aligned}$$

By the induction hypothesis for τ_2, we have that one of the following must hold:

- there are $\mathbf{v}'_2$ and v'_2 such that $\mathbb{E}[\mathbf{inject}_{\tau_2;n}\ \mathbf{v_2}] \hookrightarrow^* \mathbb{E}[\mathbf{v}'_2]$ and $\mathbb{E}[\mathsf{protect}_{\tau_2}\ \mathsf{v}_2] \hookrightarrow^* \mathbb{E}[\mathsf{v}'_2]$ for any $\mathbb{E}$ and $\mathbb{E}$ and that $(\triangleright \underline{\mathsf{W}}, \mathbf{v}'_2, \mathsf{v}'_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_\square$.
- $(\mathbb{E}[\mathbf{inject}_{\tau_2;n}\ \mathbf{v_2}], \mathbb{E}[\mathsf{protect}_{\tau_2}\ \mathsf{v}_2]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}})_\square$ for any $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$ and for any $\mathbb{E}$, $\mathbb{E}$.

In the latter case, we have by the above evaluations and by Lemma 4 that $(\mathbb{E}[\mathbf{inject}_{\tau_1\times\tau_2;n+1}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_{\tau_1\times\tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the former case, we can continue the evaluations for any $\mathbb{E}$ and for any $\mathbb{E}$ as follows:

$$\mathbb{E}[\mathbf{in}_{\times;n}\langle \mathbf{v}'_1, \mathbf{inject}_{\tau_2;n}\ \mathbf{v_2}\rangle] \hookrightarrow^* \mathbb{E}[\mathbf{in}_{\times;n}\langle \mathbf{v}'_1, \mathbf{v}'_2\rangle]$$

and

$$\mathbb{E}[\langle \mathsf{v}'_1, \mathsf{protect}_{\tau_2}\ \mathsf{v}_2\rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}'_1, \mathsf{v}'_2\rangle].$$

It remains to prove that $(\underline{\mathsf{W}}, [\mathbf{in}_{\times;n}\langle \mathbf{v}'_1, \mathbf{v}'_2\rangle], [\langle \mathsf{v}'_1, \mathsf{v}'_2\rangle]) \in \mathtt{EmulDV}_{n+1;p}$, but this follows directly by definition of $\mathtt{EmulDV}_{n+1;p}$, by the facts that $(\triangleright \underline{\mathsf{W}}, \mathbf{v}'_1, \mathsf{v}'_1) \in \triangleright \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_\square$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v}'_2, \mathsf{v}'_2) \in \triangleright \mathcal{V}[\![\mathtt{EmulDV}_{n;p}]\!]_\square$.

Now if $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{n+1;p}]\!]_\square$, then we have that one of the following cases must hold:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};n} \wedge p = \mathtt{imprecise}$

- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\mathcal{B};\mathbf{n}}(v') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$
- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\times;\mathbf{n}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \times \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$
- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \uplus \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$
- $\exists \mathbf{v}'.\, \mathbf{v} = \mathbf{in}_{\rightarrow;\mathbf{n}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \rightarrow \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$

In the first case, we know that $\square = \lesssim$ and we have that

$$\begin{aligned}
&\mathbb{E}[\mathbf{extract}_{\tau_1\times\tau_2;\mathsf{n}+1}\ \mathbf{v}] \hookrightarrow \\
&\qquad \mathbb{E}[\langle \mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.1}, \mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.2}\rangle] \hookrightarrow^* \\
&\qquad \mathbb{E}[\langle \mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathrm{omega}_{(\mathrm{UVal}_\mathsf{n}\times\mathrm{UVal}_\mathsf{n})}.\mathbf{1}, \mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.2}\rangle]
\end{aligned}$$

By definition of $\mathsf{O}(\underline{\mathsf{W}})_{\lesssim}$, we have that $(\mathbb{E}[\mathbf{extract}_{\tau_1\times\tau_2;\mathsf{n}+1}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1\times\tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})$ for any $\mathbb{E}$, $\mathbb{E}$.

We repeat the definition of $\mathtt{case}_{\times;\mathsf{n}}$ for easy reference:

$$\mathtt{case}_{\times;\mathsf{n}} = \lambda uv : \mathrm{UVal}_{\mathsf{n}+1}.\, \mathrm{case}\ uv\ \mathrm{of}\ \{\mathbf{in}_{\times;\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{(\mathrm{UVal}_\mathsf{n}\times\mathrm{UVal}_\mathsf{n})}\}$$

In the second, fourth and fifth case, we have that

$$\begin{aligned}
&\mathbb{E}[\mathbf{extract}_{\tau_1\times\tau_2;\mathsf{n}+1}\ \mathbf{v}] \hookrightarrow \\
&\qquad \mathbb{E}[\langle \mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.1}, \mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.2}\rangle] \hookrightarrow^* \\
&\qquad \mathbb{E}[\langle \mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathrm{omega}_{(\mathrm{UVal}_\mathsf{n}\times\mathrm{UVal}_\mathsf{n})}.\mathbf{1}, \mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.2}\rangle]
\end{aligned}$$

(which diverges) and for any $\mathbb{E}$ that

$$\begin{aligned}
&\mathbb{E}[\mathsf{confine}_{\tau_1\times\tau_2}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\langle \mathsf{confine}_{\tau_1}\ \mathsf{v.1}, \mathsf{confine}_{\tau_2}\ \mathsf{v.2}\rangle] \\
&\qquad\qquad \mathbb{E}[\langle \mathsf{confine}_{\tau_1}\ \mathsf{wrong}, \mathsf{confine}_{\tau_2}\ \mathsf{v.2}\rangle] \hookrightarrow \mathsf{wrong}
\end{aligned}$$

By Lemmas 4 and 6, we have that $(\mathbb{E}[\mathbf{extract}_{\tau_1\times\tau_2;\mathsf{n}+1}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1\times\tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})$ for any $\mathbb{E}$, $\mathbb{E}$.

In the third case (where $\mathbf{v} = \mathbf{in}_{\times;\mathbf{n}}(\mathbf{v}')$) we have that $\mathbf{v}' = \langle \mathbf{v_1}, \mathbf{v_2}\rangle$, $\mathsf{v} = \langle \mathsf{v}_1, \mathsf{v}_2\rangle$ with $(\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \triangleright\, \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$ and $(\underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \triangleright\, \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$, by definition of $\mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}} \times \mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$.

If $\mathsf{lev}(\underline{\mathsf{W}}) = 0$, then by Lemma 5, $(\mathbb{E}[\mathbf{extract}_{\tau_1\times\tau_2;\mathsf{n}}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1\times\tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_{\square}$ for any $\mathbb{E}$, $\mathbb{E}$.

If $\mathsf{lev}(\underline{\mathsf{W}}) > 0$, then we have that $(\triangleright\, \underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$ and $(\triangleright\, \underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_{\square}$.

We already have for any $\mathbb{E}$ that

$$\begin{aligned}
&\mathbb{E}[\mathbf{extract}_{\tau_1\times\tau_2;\mathsf{n}+1}\ \mathbf{v}] \hookrightarrow \\
&\qquad \mathbb{E}[\langle \mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.1}, \mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.2}\rangle] \hookrightarrow \\
&\mathbb{E}\left[\left\langle \begin{array}{l}\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathrm{case}\ \mathbf{v}\ \mathrm{of}\ \{\mathbf{in}_{\times;\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ _ \mapsto \mathrm{omega}_{(\mathrm{UVal}_\mathsf{n}\times\mathrm{UVal}_\mathsf{n})}\}.\mathbf{1}, \\ \qquad \mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.2}\end{array}\right\rangle\right] \hookrightarrow \\
&\qquad \mathbb{E}[\langle \mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{v}'.\mathbf{1}, \mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.2}\rangle] \hookrightarrow \\
&\qquad\qquad \mathbb{E}[\langle \mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{v_1}, \mathbf{extract}_{\tau_2;\mathsf{n}}\ \mathtt{case}_{\times;\mathsf{n}}\ \mathbf{v.2}\rangle]
\end{aligned}$$

and for any $\mathbb{E}$ that

$$\mathbb{E}[\mathsf{confine}_{\tau_1 \times \tau_2}\ \mathsf{v}] \hookrightarrow \mathbb{E}[\langle \mathsf{confine}_{\tau_1}\ \mathsf{v}.1, \mathsf{confine}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow \\ \mathbb{E}[\langle \mathsf{confine}_{\tau_1}\ \mathsf{v}_1, \mathsf{confine}_{\tau_2}\ \mathsf{v}.2\rangle]$$

By induction, we know that one of the following cases holds:

- there exist $\mathbf{v}_1'$ and v_1' such that $\mathbb{E}[\mathbf{extract}_{\tau_1;n}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v}_1']$ and $\mathbb{E}[\mathsf{confine}_{\tau_1}\ \mathsf{v}_1] \hookrightarrow^* \mathbb{E}[\mathsf{v}_1']$ for any $\mathbb{E}$ and $\mathbb{E}$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v}_1', \mathsf{v}_1') \in \mathcal{V}[\![\tau_1]\!]_\square$
- $(\mathbb{E}[\mathbf{extract}_{\tau_1;n}\ \mathbf{v_1}], \mathbb{E}[\mathsf{confine}_{\tau_1}\ \mathsf{v}_1]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the latter case, by Lemma 4 and the above evaluation, we get that $(\mathbb{E}[\mathbf{extract}_{\tau_1 \times \tau_2;n}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1 \times \tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the former case, the above evaluation judgements continue as follows for any $\mathbb{E}$ and $\mathbb{E}$:

$$\mathbb{E}[\langle \mathbf{extract}_{\tau_1;n}\ \mathbf{v_1}, \mathbf{extract}_{\tau_2;n}\ \mathsf{case}_{\times;n}\ \mathbf{v.2}\rangle] \hookrightarrow^* \\ \mathbb{E}[\langle \mathbf{v}_1', \mathbf{extract}_{\tau_2;n}\ \mathsf{case}_{\times;n}\ \mathbf{v.2}\rangle] \hookrightarrow \\ \mathbb{E}[\langle \mathbf{v}_1', \mathbf{extract}_{\tau_2;n}\ \text{case}\ \mathbf{v}\ \text{of}\ \{\mathbf{in}_{\times;n}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \text{omega}_{(\text{UVal}_n \times \text{UVal}_n)}\}.\mathbf{2}\rangle] \hookrightarrow \\ \mathbb{E}[\langle \mathbf{v}_1', \mathbf{extract}_{\tau_2;n}\ \mathbf{v}'.\mathbf{2}\rangle] \hookrightarrow \mathbb{E}[\langle \mathbf{v}_1', \mathbf{extract}_{\tau_2;n}\ \mathbf{v_2}\rangle]$$

and

$$\mathbb{E}[\langle \mathsf{confine}_{\tau_1}\ \mathsf{v}_1, \mathsf{confine}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow^* \\ \mathbb{E}[\langle \mathsf{v}_1', \mathsf{confine}_{\tau_2}\ \mathsf{v}.2\rangle] \hookrightarrow \\ \mathbb{E}[\langle \mathsf{v}_1', \mathsf{confine}_{\tau_2}\ \mathsf{v}_2\rangle]$$

Again by induction, we know that one of the following cases holds:

- there exist $\mathbf{v}_2'$ and v_2' such that $\mathbb{E}[\mathbf{extract}_{\tau_2;n}\ \mathbf{v_2}] \hookrightarrow^* \mathbb{E}[\mathbf{v}_2']$ and $\mathbb{E}[\mathsf{confine}_{\tau_2}\ \mathsf{v}_2] \hookrightarrow^* \mathbb{E}[\mathsf{v}_2']$ for any $\mathbb{E}$ and $\mathbb{E}$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v}_2', \mathsf{v}_2') \in \mathcal{V}[\![\tau_2]\!]_\square$.
- $(\mathbb{E}[\mathbf{extract}_{\tau_2;n}\ \mathbf{v_2}], \mathbb{E}[\mathsf{confine}_{\tau_2}\ \mathsf{v}_2]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the latter case, by Lemma 4 and the above (continued) evaluation, we get that $(\mathbb{E}[\mathbf{extract}_{\tau_1 \times \tau_2;n}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1 \times \tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the former case, the evaluation judgements continue further as follows for any $\mathbb{E}$ and $\mathbb{E}$:

$$\mathbb{E}[\langle \mathbf{v}_1', \mathbf{extract}_{\tau_2;n}\ \mathbf{v_2}\rangle] \hookrightarrow^* \mathbb{E}[\langle \mathbf{v}_1', \mathbf{v}_2'\rangle]$$

and

$$\mathbb{E}[\langle \mathsf{v}_1', \mathsf{confine}_{\tau_2}\ \mathsf{v}_2\rangle] \hookrightarrow^* \mathbb{E}[\langle \mathsf{v}_1', \mathsf{v}_2'\rangle]$$

It now suffices to prove that $(\underline{\mathsf{W}}, \langle \mathbf{v}_1', \mathbf{v}_2'\rangle, \langle \mathsf{v}_1', \mathsf{v}_2'\rangle) \in \mathcal{V}[\![\tau_1 \times \tau_2]\!]_\square$, but this follows directly from $(\triangleright \underline{\mathsf{W}}, \mathbf{v}_1', \mathsf{v}_1') \in \mathcal{V}[\![\tau_1]\!]_\square$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v}_2', \mathsf{v}_2') \in \mathcal{V}[\![\tau_2]\!]_\square$.

- $\tau = \tau_1 \uplus \tau_2$: We have that

$$\mathbf{inject}_{\tau_1\uplus\tau_2;\mathsf{n}+1} = \lambda \mathbf{v} : \tau_1 \uplus \tau_2.\, \mathbf{in}_{\uplus;\mathbf{n}} \left(\text{case } \mathbf{v} \text{ of } \left| \begin{array}{l} \text{inl } x \to \text{inl } (\mathbf{inject}_{\tau_1;\mathsf{n}}\ x) \\ \text{inr } x \to \text{inr } (\mathbf{inject}_{\tau_2;\mathsf{n}}\ x) \end{array} \right. \right)$$

$$\mathbf{extract}_{\tau_1\uplus\tau_2;\mathsf{n}+1} = \lambda uv : \text{UVal}_{\mathsf{n}+1}.\, \text{case } \mathtt{case}_{\uplus;\mathbf{n}}\ uv \text{ of } \left| \begin{array}{l} \text{inl } x \to \text{inl } (\mathbf{extract}_{\tau_1;\mathsf{n}}\ x) \\ \text{inr } x \to \text{inr } (\mathbf{extract}_{\tau_2;\mathsf{n}}\ x) \end{array} \right.$$

$$\mathsf{protect}_{\tau_1\uplus\tau_2} = \lambda \mathsf{y}.\, \text{case } \mathsf{y} \text{ of inl } \mathsf{x} \to \text{inl } (\mathsf{protect}_{\tau_1}\ \mathsf{x}) \mid \text{inr } \mathsf{x} \to \text{inr } (\mathsf{protect}_{\tau_2}\ \mathsf{x})$$

$$\mathsf{confine}_{\tau_1\uplus\tau_2} \stackrel{\mathsf{def}}{=} \lambda \mathsf{y}.\, \text{case } \mathsf{y} \text{ of inl } \mathsf{x} \to \text{inl } (\mathsf{confine}_{\tau_1}\ \mathsf{x}) \mid \text{inr } \mathsf{x} \to \text{inr } (\mathsf{confine}_{\tau_2}\ \mathsf{x})$$

If $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau_1 \uplus \tau_2]\!]_\square$, then we have that either $\mathbf{v} = \text{inl } \mathbf{v_1}$ and $\mathsf{v} = \text{inl } \mathsf{v}_1$ for some $\mathbf{v_1}$, v_1 with $(\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \triangleright \mathcal{V}[\![\tau_1]\!]_\square$ or $\mathbf{v} = \text{inr } \mathbf{v_2}$ and $\mathsf{v} = \text{inr } \mathsf{v}_2$ for some $\mathbf{v_2}$, v_2 with $(\underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \triangleright \mathcal{V}[\![\tau_2]\!]_\square$. We prove the result for the first case, the other case is completely similar.

If $\mathsf{lev}(\underline{\mathsf{W}}) = 0$, then we know by Lemma 5 that $(\mathbb{E}[\mathbf{inject}_{\tau;\mathsf{n}}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_\tau\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$. If $\mathsf{lev}(\underline{\mathsf{W}}) > 0$, then we have that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\tau_1]\!]_\square$.

We have for any $\mathbb{E}$ that

$$\begin{array}{l} \mathbb{E}[\mathbf{inject}_{\tau_1\uplus\tau_2;\mathsf{n}+1}\ \mathbf{v}] \hookrightarrow \\ \mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n}}\ (\text{case } \mathbf{v} \text{ of inl } \mathbf{x} \to \text{inl } (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{x}) \mid \text{inr } \mathbf{x} \to \text{inr } (\mathbf{inject}_{\tau_2;\mathsf{n}}\ \mathbf{x}))] \hookrightarrow \\ \qquad \mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n}}\ (\text{inl } (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_1}))] \end{array}$$

and for any $\mathbb{E}$ that

$$\begin{array}{l} \mathbb{E}[\mathsf{protect}_{\tau_1\uplus\tau_2}\ \mathsf{v}] \hookrightarrow \\ \quad \mathbb{E}[\text{case } \mathsf{v} \text{ of inl } \mathsf{x} \to \text{inl } (\mathsf{protect}_{\tau_1}\ \mathsf{x}) \mid \text{inr } \mathsf{x} \to \text{inr } (\mathsf{protect}_{\tau_2}\ \mathsf{x})] \hookrightarrow \\ \qquad \mathbb{E}[\text{inl } (\mathsf{protect}_{\tau_1}\ \mathsf{v}_1)] \end{array}$$

By induction, we know that one of the following cases must hold:

- there are $\mathbf{v}'_1$ and v'_1 such that $\mathbb{E}[\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v}'_1]$ and $\mathbb{E}[\mathsf{protect}_{\tau_1}\ \mathsf{v}_1] \hookrightarrow^* \mathbb{E}[\mathsf{v}'_1]$ for any $\mathbb{E}$ and $\mathbb{E}$ and that $(\triangleright \underline{\mathsf{W}}, \mathbf{v}'_1, \mathsf{v}'_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_\square$.
- $(\mathbb{E}[\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_1}], \mathbb{E}[\mathsf{protect}_{\tau_1}\ \mathsf{v}_1]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}})_\square$ for all $\mathbb{E}$ and $\mathbb{E}$.

In the latter case, it follows by the above evaluation and by Lemma 4 that $(\mathbb{E}[\mathbf{inject}_{\tau_1\uplus\tau_2;\mathsf{n}+1}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_{\tau_1\uplus\tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for all $\mathbb{E}$ and $\mathbb{E}$.

In the former case, we have for any $\mathbb{E}$ that

$$\mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n}}\ (\text{inl } (\mathbf{inject}_{\tau_1;\mathsf{n}}\ \mathbf{v_1}))] \hookrightarrow^* \mathbb{E}[\mathbf{in}_{\uplus;\mathbf{n}}\ (\text{inl } \mathbf{v}'_1)]$$

and for any $\mathbb{E}$ that

$$\mathbb{E}[\text{inl } (\mathsf{protect}_{\tau_1}\ \mathsf{v}_1)] \hookrightarrow^* \mathbb{E}[\text{inl } \mathsf{v}'_1]$$

It remains to prove that $(\underline{\mathsf{W}}, [\mathbf{in}_{\uplus;\mathbf{n}}\ (\mathrm{inl}\ \mathbf{v_1'})], [\mathrm{inl}\ \mathsf{v}_1']) \in \mathtt{EmulDV}_{\mathsf{n+1;p}}$, but this follows directly by definition of $\mathtt{EmulDV}_{\mathsf{n+1;p}}$, $\mathcal{V}[\![\tau_1 \uplus \tau_2]\!]_\square$ and by the fact that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1'}, \mathsf{v}_1') \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_\square$.

Now if $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n+1;p}}]\!]_\square$, then we have that one of the following cases must hold:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};\mathbf{n}} \wedge p = \mathtt{imprecise}$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\mathcal{B};\mathbf{n}}(v') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_\square$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\times;\mathbf{n}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \times \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_\square$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \uplus \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_\square$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\to;\mathbf{n}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \to \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_\square$

We repeat the definition of $\mathtt{case}_{\uplus;\mathbf{n}}$ for easy reference:

$$\mathtt{case}_{\uplus;\mathbf{n}} = \lambda uv : \mathrm{UVal}_{\mathsf{n+1}}.\ \mathrm{case}\ uv\ \mathrm{of}\ \{\mathbf{in}_{\uplus;\mathbf{n}}\ \mathbf{x} \mapsto \mathbf{x};\ \_ \mapsto \mathrm{omega}_{(\mathrm{UVal}_{\mathsf{n}} \uplus \mathrm{UVal}_{\mathsf{n}})}\}$$

In the first case, we know that $\square = \lesssim$ and

$$\mathbb{E}[\mathbf{extract}_{\tau_1 \uplus \tau_2;\mathsf{n+1}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathrm{case\ omega}_{(\mathrm{UVal}_{\mathsf{n}} \uplus \mathrm{UVal}_{\mathsf{n}})}\ \mathrm{of}\ \begin{array}{l} \mathrm{inl}\ x \to \mathrm{inl}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ x) \\ \mathrm{inr}\ x \to \mathrm{inr}\ (\mathbf{extract}_{\tau_2;\mathsf{n}}\ x) \end{array}]$$

which diverges. By definition of $\mathsf{O}(\underline{\mathsf{W}})_{\lesssim}$, we know that $(\mathbb{E}[\mathbf{extract}_{\tau_1 \uplus \tau_2;\mathsf{n+1}}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1 \uplus \tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})$ for any $\mathbb{E}$, $\mathbb{E}$.

In the second, third and fifth case, we have for any $\mathbb{E}$ that

$$\mathbb{E}[\mathbf{extract}_{\tau_1 \uplus \tau_2;\mathsf{n+1}}\ \mathbf{v}] \hookrightarrow \mathbb{E}[\mathrm{case\ omega}_{(\mathrm{UVal}_{\mathsf{n}} \uplus \mathrm{UVal}_{\mathsf{n}})}\ \mathrm{of}\ \begin{array}{l} \mathrm{inl}\ x \to \mathrm{inl}\ (\mathbf{extract}_{\tau_1;\mathsf{n}}\ x) \\ \mathrm{inr}\ x \to \mathrm{inr}\ (\mathbf{extract}_{\tau_2;\mathsf{n}}\ x) \end{array}]$$

(which diverges) and for any $\mathbb{E}$ that

$$\begin{array}{l} \mathbb{E}[\mathsf{confine}_{\tau_1 \uplus \tau_2}\ \mathsf{v}] \hookrightarrow \\ \quad \mathbb{E}[\mathrm{case}\ \mathsf{v}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x} \to \mathrm{inl}\ (\mathsf{confine}_{\tau_1}\ \mathsf{x}) \mid \mathrm{inr}\ \mathsf{x} \to \mathrm{inr}\ (\mathsf{confine}_{\tau_2}\ \mathsf{x})] \hookrightarrow \\ \qquad\qquad\qquad\qquad\qquad\qquad \mathbb{E}[\mathsf{wrong}] \hookrightarrow \mathsf{wrong} \end{array}$$

By Lemmas 4 and 6, we have that $(\mathbb{E}[\mathbf{extract}_{\tau_1 \uplus \tau_2;\mathsf{n+1}}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1 \uplus \tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})$ for any $\mathbb{E}$, $\mathbb{E}$.

In the fourth case (where $\mathbf{v} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathbf{v}')$) we have by definition of $\mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \uplus \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_\square$ that either $\mathbf{v}' = \mathrm{inl}\ \mathbf{v_1}$, $\mathsf{v} = \mathrm{inl}\ \mathsf{v}_1$ with $(\underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \triangleright \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_\square$, or $\mathbf{v}' = \mathrm{inr}\ \mathbf{v_2}$, $\mathsf{v} = \mathrm{inr}\ \mathsf{v}_2$ with $(\underline{\mathsf{W}}, \mathbf{v_2}, \mathsf{v}_2) \in \triangleright \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}}]\!]_\square$. We prove the result for the first case, the other case is completely similar.

If $\mathsf{lev}(\underline{\mathsf{W}}) = 0$, then we know by Lemma 5 that $(\mathbb{E}[\mathbf{extract}_{\tau_1 \uplus \tau_2;\mathsf{n}}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1 \uplus \tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$. If $\mathsf{lev}(\underline{\mathsf{W}}) > 0$, then we have that $(\triangleright \underline{\mathsf{W}}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n};\mathsf{p}}]\!]_\square$.

We then already have for any $\mathbb{E}$ that

$$
\begin{aligned}
&\mathbb{E}[\mathbf{extract}_{\tau_1 \uplus \tau_2;\mathsf{n}+1}\ \mathbf{v}] \hookrightarrow \\
&\qquad \mathbb{E}[\text{case } \mathtt{case}_{\uplus;\mathsf{n}}\ \mathbf{v} \text{ of } \left| \begin{array}{l} \text{inl } x \to \text{inl } (\mathbf{extract}_{\tau_1;\mathsf{n}}\ x) \\ \text{inr } x \to \text{inr } (\mathbf{extract}_{\tau_2;\mathsf{n}}\ x) \end{array} \right.] \hookrightarrow \\
&\qquad\quad \mathbb{E}[\text{case } \mathbf{v}' \text{ of } \left| \begin{array}{l} \text{inl } x \to \text{inl } (\mathbf{extract}_{\tau_1;\mathsf{n}}\ x) \\ \text{inr } x \to \text{inr } (\mathbf{extract}_{\tau_2;\mathsf{n}}\ x) \end{array} \right.] \hookrightarrow \\
&\qquad\qquad\qquad\qquad\qquad\qquad \mathbb{E}[\text{inl } (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{v_1})]
\end{aligned}
$$

and for any $\mathbb{E}$ that

$$
\begin{aligned}
&\mathbb{E}[\mathsf{confine}_{\tau_1 \uplus \tau_2}\ \mathsf{v}] \hookrightarrow \\
&\quad \mathbb{E}[\text{case } \mathsf{v} \text{ of inl } \mathsf{x} \to \text{inl } (\mathsf{confine}_{\tau_1}\ \mathsf{x}) \mid \text{inr } \mathsf{x} \to \text{inr } (\mathsf{confine}_{\tau_2}\ \mathsf{x})] \hookrightarrow \\
&\qquad\qquad\qquad\qquad\qquad\qquad \mathbb{E}[\text{inl } (\mathsf{confine}_{\tau_1}\ \mathsf{x})]
\end{aligned}
$$

By induction, we know that one of the following cases holds:

- there exist $\mathbf{v}'_1$ and v'_1 such that $\mathbb{E}[\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{v_1}] \hookrightarrow^* \mathbb{E}[\mathbf{v}'_1]$ and $\mathbb{E}[\mathsf{confine}_{\tau_1}\ \mathsf{v}_1] \hookrightarrow^* \mathbb{E}[\mathsf{v}'_1]$ for any $\mathbb{E}$ and $\mathbb{E}$ and $(\triangleright \underline{\mathsf{W}}, \mathbf{v}'_1, \mathsf{v}'_1) \in \mathcal{V}[\![\tau_1]\!]_\square$
- $(\mathbb{E}[\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{v_1}], \mathbb{E}[\mathsf{confine}_{\tau_1}\ \mathsf{v}_1]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the latter case, by Lemma 4 and the above evaluation, we get that $(\mathbb{E}[\mathbf{extract}_{\tau_1 \uplus \tau_2;\mathsf{n}}\ \mathbf{v}], \mathbb{E}[\mathsf{confine}_{\tau_1 \uplus \tau_2}\ \mathsf{v}]) \in \mathsf{O}(\underline{\mathsf{W}})_\square$ for any $\mathbb{E}$, $\mathbb{E}$.

In the former case, the above evaluation judgements continue as follows for any $\mathbb{E}$ and $\mathbb{E}$:

$$\mathbb{E}[\text{inl } (\mathbf{extract}_{\tau_1;\mathsf{n}}\ \mathbf{v_1})] \hookrightarrow^* \mathbb{E}[\text{inl } \mathbf{v}'_1]$$

and

$$\mathbb{E}[\text{inl } (\mathsf{confine}_{\tau_1}\ \mathsf{x})] \hookrightarrow^* \mathbb{E}[\text{inl } \mathsf{v}'_1]$$

It now suffices to prove that $(\underline{\mathsf{W}}, \text{inl } \mathbf{v}'_1, \text{inl } \mathsf{v}'_1) \in \mathcal{V}[\![\tau_1 \uplus \tau_2]\!]_\square$, but this follows directly from $(\triangleright \underline{\mathsf{W}}, \mathbf{v}'_1, \mathsf{v}'_1) \in \mathcal{V}[\![\tau_1]\!]_\square$.

$\square$

Theorem 10 (Inject is protect and extract is confine). *If ($m \geq n$ and $p =$* `precise`*) or ($\square = \lesssim$ and $p =$* `imprecise`*) and if $\mathbf{\Gamma} \vdash \mathbf{t}\ \square_\mathsf{n}\ \mathsf{t} : \tau$, then*

$$\mathbf{\Gamma} \vdash \mathbf{inject}_{\tau;\mathsf{m}}\ \mathbf{t}\ \square_\mathsf{n}\ \mathsf{protect}_\tau\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m};\mathsf{p}}.$$

If ($m \geq n$ and $p =$ `precise`*) or ($\square = \lesssim$ and $p =$* `imprecise`*) and if $\mathbf{\Gamma} \vdash \mathbf{t}\ \square_\mathsf{n}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m};\mathsf{p}}$ then*

$$\mathbf{\Gamma} \vdash \mathbf{extract}_{\tau;\mathsf{m}}\ \mathbf{t}\ \square_\mathsf{n}\ \mathsf{confine}_\tau\ \mathsf{t} : \tau.$$

Proof. Take $\underline{W}$ with $\mathsf{lev}(\underline{W}) \leq n$. Take $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\Gamma]\!]_{\square}$. Then we need to show that

$$(\underline{W}, \mathbf{inject}_{\tau;m}\ \mathbf{t}\gamma, \mathsf{protect}_{\tau}\ \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{m;p}]\!]_{\square}.$$

We know that $(\underline{W}, \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\tau]\!]_{\square}$. By Lemma 19, it then suffices to show that for all $\underline{W}' \sqsupseteq \underline{W}$, $(\underline{W}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau]\!]_{\square}$, we have that

$$(\underline{W}, \mathbf{inject}_{\tau;m}\ \mathbf{v}, \mathsf{protect}_{\tau}\ \mathsf{v}) \in \mathcal{E}[\![\mathtt{EmulDV}_{m;p}]\!]_{\square}.$$

So, take $(\underline{W}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\mathtt{EmulDV}_{m;p}]\!]_{\square}$. Then we need to show that

$$(\mathbb{E}[\mathbf{inject}_{\tau;m}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_{\tau}\ \mathsf{v}]) \in \mathsf{O}(\underline{W}).$$

By Lemma 40, we get that one of the following cases must hold:

- $\mathbf{v}'$ and v' such that $\mathbb{E}[\mathbf{inject}_{\tau;m}\ \mathbf{v}] \hookrightarrow^* \mathbb{E}[\mathbf{v}']$ and $\mathbb{E}[\mathsf{protect}_{\tau}\ \mathsf{v}] \hookrightarrow^* \mathbb{E}[\mathsf{v}']$ and $(\underline{W}, \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\mathtt{EmulDV}_{m;p}]\!]_{\square}$. By Lemma 4, it suffices to prove that

  $$(\mathbb{E}[\mathbf{v}'], \mathbb{E}[\mathsf{v}']) \in \mathsf{O}(\underline{W}).$$

  But this follows directly from $(\underline{W}, \mathbf{v}', \mathsf{v}') \in \mathcal{V}[\![\mathtt{EmulDV}_{m;p}]\!]_{\square}$ and $(\underline{W}, \mathbb{E}, \mathbb{E}) \in \mathcal{K}[\![\mathtt{EmulDV}_{m;p}]\!]_{\square}$.

- $(\mathbb{E}[\mathbf{inject}_{\tau;m}\ \mathbf{v}], \mathbb{E}[\mathsf{protect}_{\tau}\ \mathsf{v}]) \in \mathsf{O}(\underline{W})_{\square}$ for any $\mathbb{E}$, $\mathbb{E}$. The result follows directly by definition of $\mathcal{E}[\![\mathtt{EmulDV}_{m;p}]\!]_{\square}$.

$\square$

6.5 Emulating λ^{u} in UVal

$$\begin{aligned}
\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}) &: \mathrm{UVal}_{\mathsf{n}}\\
\mathrm{emulate}_{\mathsf{n}}(\mathtt{unit}) &\stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\mathtt{Unit};\mathbf{n}}\ \mathtt{unit})\\
\mathrm{emulate}_{\mathsf{n}}(\mathtt{true}) &\stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ \mathtt{true})\\
\mathrm{emulate}_{\mathsf{n}}(\mathtt{false}) &\stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\mathtt{Bool};\mathbf{n}}\ \mathtt{false})\\
\mathrm{emulate}_{\mathsf{n}}(\mathsf{x}) &\stackrel{\mathsf{def}}{=} \mathbf{x}\\
\mathrm{emulate}_{\mathsf{n}}(\lambda \mathsf{x}.\, \mathsf{t}) &\stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\to;\mathbf{n}}\ (\lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{emulate}_{\mathsf{n}}(\mathsf{t})))\\
\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1\ \mathsf{t}_2) &\stackrel{\mathsf{def}}{=} \mathtt{case}_{\to;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)))\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2)\\
\mathrm{emulate}_{\mathsf{n}}(\langle \mathsf{t}_1, \mathsf{t}_2 \rangle) &\stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\times;\mathbf{n}}\ \langle \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1), \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2) \rangle)\\
\mathrm{emulate}_{\mathsf{n}}(\mathrm{inl}\ \mathsf{t}) &\stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\uplus;\mathbf{n}}\ (\mathrm{inl}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t})))\\
\mathrm{emulate}_{\mathsf{n}}(\mathrm{inr}\ \mathsf{t}) &\stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\uplus;\mathbf{n}}\ (\mathrm{inr}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t})))\\
\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}.1) &\stackrel{\mathsf{def}}{=} (\mathtt{case}_{\times;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathsf{t})))).\mathbf{1}\\
\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}.2) &\stackrel{\mathsf{def}}{=} (\mathtt{case}_{\times;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathsf{t})))).\mathbf{2}\\
\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}; \mathsf{t}') &\stackrel{\mathsf{def}}{=} (\mathtt{case}_{\mathtt{Unit};\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}(\mathsf{t})))); \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}')\\
\mathrm{emulate}_{\mathsf{n}}(\mathsf{wrong}) &\stackrel{\mathsf{def}}{=} \mathrm{omega}
\end{aligned}$$

$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{case}\ \mathsf{t}_1\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x} \mapsto \mathsf{t}_2 \mid \mathrm{inr}\ \mathsf{x} \mapsto \mathsf{t}_3) \stackrel{\mathsf{def}}{=}$$
$$\mathrm{case}\ \mathtt{case}_{\uplus;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)))\ \mathrm{of}$$
$$\mathrm{inl}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2) \mid \mathrm{inr}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_3)$$

$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{if}\ \mathsf{t}\ \mathrm{then}\ \mathsf{t}_1\ \mathrm{else}\ \mathsf{t}_2) \stackrel{\mathsf{def}}{=}$$
$$\mathrm{if}\ (\mathtt{case}_{\mathtt{Bool};\mathtt{n}}(\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}\mathsf{t})))\ \mathrm{then}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)\ \mathrm{else}$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2)$$

$$\mathrm{emulate}_{\mathsf{n}}(\cdot) \stackrel{\mathsf{def}}{=} \cdot$$
$$\mathrm{emulate}_{\mathsf{n}}(\lambda \mathsf{x}.\, \mathfrak{C}) \stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\to;\mathbf{n}}\ (\lambda \mathbf{x} : \mathrm{UVal}_{\mathsf{n}}.\, \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})))$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C}\ \mathsf{t}_2) \stackrel{\mathsf{def}}{=} \mathtt{case}_{\to;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})))\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2)$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1\ \mathfrak{C}) \stackrel{\mathsf{def}}{=} \mathtt{case}_{\to;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)))\ \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C}.1) \stackrel{\mathsf{def}}{=} (\mathtt{case}_{\times;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})))).\mathbf{1}$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C}.2) \stackrel{\mathsf{def}}{=} (\mathtt{case}_{\times;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})))).\mathbf{2}$$
$$\mathrm{emulate}_{\mathsf{n}}(\langle \mathfrak{C}, \mathsf{t}_2\rangle) \stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\times;\mathbf{n}}\ \langle \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C}), \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2)\rangle)$$
$$\mathrm{emulate}_{\mathsf{n}}(\langle \mathsf{t}_1, \mathfrak{C}\rangle) \stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\times;\mathbf{n}}\ \langle \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1), \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})\rangle)$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{inl}\ \mathfrak{C}) \stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\uplus;\mathbf{n}}\ (\mathrm{inl}\ \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})))$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{inr}\ \mathfrak{C}) \stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{n};1}\ (\mathbf{in}_{\uplus;\mathbf{n}}\ (\mathrm{inr}\ \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})))$$

$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{case}\ \mathfrak{C}\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x} \mapsto \mathsf{t}_2 \mid \mathrm{inr}\ \mathsf{x} \mapsto \mathsf{t}_3) \stackrel{\mathsf{def}}{=}$$
$$\mathrm{case}\ \mathtt{case}_{\uplus;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})))\ \mathrm{of}$$
$$\mathrm{inl}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2) \mid \mathrm{inr}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_3)$$

$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{case}\ \mathsf{t}_1\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x} \mapsto \mathfrak{C} \mid \mathrm{inr}\ \mathsf{x} \mapsto \mathsf{t}_3) \stackrel{\mathsf{def}}{=}$$
$$\mathrm{case}\ \mathtt{case}_{\uplus;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)))\ \mathrm{of}$$
$$\mathrm{inl}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C}) \mid \mathrm{inr}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_3)$$

$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{case}\ \mathsf{t}_1\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x} \mapsto \mathsf{t}_2 \mid \mathrm{inr}\ \mathsf{x} \mapsto \mathfrak{C}) \stackrel{\mathsf{def}}{=}$$
$$\mathrm{case}\ \mathtt{case}_{\uplus;\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}\ (\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)))\ \mathrm{of}$$
$$\mathrm{inl}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2) \mid \mathrm{inr}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})$$

$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{if}\ \mathfrak{C}\ \mathrm{then}\ \mathsf{t}_1\ \mathrm{else}\ \mathsf{t}_2) \stackrel{\mathsf{def}}{=} \begin{array}{l}\mathrm{if}\ (\mathtt{case}_{\mathtt{Bool};\mathtt{n}}(\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C}))))\\ \qquad \mathrm{then}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)\ \mathrm{else}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2)\end{array}$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{if}\ \mathsf{t}\ \mathrm{then}\ \mathfrak{C}\ \mathrm{else}\ \mathsf{t}_2) \stackrel{\mathsf{def}}{=} \begin{array}{l}\mathrm{if}\ (\mathtt{case}_{\mathtt{Bool};\mathtt{n}}(\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}))))\\ \qquad \mathrm{then}\ \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})\ \mathrm{else}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2)\end{array}$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathrm{if}\ \mathsf{t}\ \mathrm{then}\ \mathsf{t}_1\ \mathrm{else}\ \mathfrak{C}) \stackrel{\mathsf{def}}{=} \begin{array}{l}\mathrm{if}\ (\mathtt{case}_{\mathtt{Bool};\mathtt{n}}(\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}))))\\ \qquad \mathrm{then}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)\ \mathrm{else}\ \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})\end{array}$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C}; \mathsf{t}') \stackrel{\mathsf{def}}{=} (\mathtt{case}_{\mathtt{Unit};\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})))); \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}')$$
$$\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}; \mathfrak{C}) \stackrel{\mathsf{def}}{=} (\mathtt{case}_{\mathtt{Unit};\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}(\mathsf{t})))); \mathrm{emulate}_{\mathsf{n}}(\mathfrak{C})$$

Lemma 41 (Compatibility lemma of emulation for lambda)**.** *If ($m > n$ and $p =$ $\texttt{precise}$) or ($\square = \lesssim$ and $p = \texttt{imprecise}$), then we have that if* $\texttt{toEmul}(\Gamma, x)_{m;p} \vdash \mathbf{t} \ \square_n \ t : \texttt{EmulDV}_{m;p}$*, then*

$$\texttt{toEmul}(\Gamma)_{m;p} \vdash \text{downgrade}_{m;1} \ (\mathbf{in}_{\to;m} \ (\lambda\mathbf{x} : \text{UVal}_m.\, \mathbf{t})) \ \square_n \ \lambda x.t : \texttt{EmulDV}_{m;p}.$$

Proof. By Theorem 9, it suffices to prove that

$$\texttt{toEmul}(\Gamma)_{m;p} \vdash \mathbf{in}_{\to;m} \ (\lambda\mathbf{x} : \text{UVal}_m.\, \mathbf{t}) \ \square_n \ \lambda x.t : \texttt{EmulDV}_{m+1;p}.$$

Take $\underline{W}$ such that $\mathsf{lev}(\underline{W}) \leq n$ and $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\texttt{toEmul}(\Gamma)_{m;p}]\!]_\square$. Then we need to show that

$$(\underline{W}, \mathbf{in}_{\to;m} \ (\lambda\mathbf{x} : \text{UVal}_m.\, \mathbf{t})\gamma, \lambda x.t\gamma) \in \mathcal{E}[\![\texttt{EmulDV}_{m+1;p}]\!]_\square,$$

or (by Lemma 10)

$$(\underline{W}, \mathbf{in}_{\to;m} \ (\lambda\mathbf{x} : \text{UVal}_m.\, \mathbf{t}\gamma), \lambda x.t\gamma) \in \mathcal{V}[\![\texttt{EmulDV}_{m+1;p}]\!]_\square.$$

By definition of $\mathcal{V}[\![\texttt{EmulDV}_{m+1;p}]\!]_\square$, it suffices to prove that $\lambda\mathbf{x} : \text{UVal}_\mathbf{m}.\, \mathbf{t}\gamma$ is in $\mathbf{oftype}(\texttt{EmulDV}_{m;p} \to \texttt{EmulDV}_{m;p})$, which holds since $\mathbf{t}$ is well-typed and

$$(\underline{W}, \lambda\mathbf{x} : \text{UVal}_\mathbf{m}.\, \mathbf{t}\gamma, \lambda x.t\gamma) \in \mathcal{V}[\![\texttt{EmulDV}_{m;p} \to \texttt{EmulDV}_{m;p}]\!]_\square.$$

So, take $\underline{W}' \sqsupset \underline{W}$ and $(\underline{W}', \mathbf{v}, v) \in \mathcal{V}[\![\texttt{EmulDV}_{m;p}]\!]_\square$. We then need to prove that

$$(\underline{W}', \mathbf{t}\gamma[\mathbf{v}/\mathbf{x}], t\gamma[v/x]) \in \mathcal{E}[\![\texttt{EmulDV}_{m;p}]\!]_\square.$$

By Lemma 11, we get that $(\underline{W}', \gamma, \gamma) \in \mathcal{G}[\![\texttt{toEmul}(\Gamma)_{m;p}]\!]_\square$. If we combine this with $(\underline{W}', \mathbf{v}, v) \in \mathcal{V}[\![\texttt{EmulDV}_{m;p}]\!]_\square$, then we get that $(\underline{W}', \gamma[\mathbf{x} \mapsto \mathbf{v}], \gamma[x \mapsto v]) \in \mathcal{G}[\![\texttt{toEmul}(\Gamma, x)_{m;p}]\!]_\square$.

Since $\mathsf{lev}(\underline{W}') < \mathsf{lev}(\underline{W}) \leq n$, we have that $\mathsf{lev}(\underline{W}') \leq n$. It now follows from $\texttt{toEmul}(\Gamma, x)_{m;p} \vdash \mathbf{t} \ \square_n \ t : \texttt{EmulDV}_{m;p}$ that

$$(\underline{W}', \mathbf{t}\gamma[\mathbf{v}/\mathbf{x}], t\gamma[v/x]) \in \mathcal{E}[\![\texttt{EmulDV}_{m;p}]\!]_\square,$$

as required. ∎

Lemma 42 (Compatibility lemma of emulation for application)**.** *If ($m > n$ and $p = \texttt{precise}$) or ($\square = \lesssim$ and $p = \texttt{imprecise}$), then we have that if* $\texttt{toEmul}(\Gamma)_{m;p} \vdash \mathbf{t_1} \ \square_n \ t_1 : \texttt{EmulDV}_{m;p}$*, and if* $\texttt{toEmul}(\Gamma)_{m;p} \vdash \mathbf{t_2} \ \square_n \ t_2 : \texttt{EmulDV}_{m;p}$*, then*

$$\texttt{toEmul}(\Gamma)_{m;p} \vdash \texttt{case}_{\to;m} \ (\text{upgrade}_{m;1} \ \mathbf{t_1}) \ \mathbf{t_2} \ \square_n \ t_1 \ t_2 : \texttt{EmulDV}_{m;p}.$$

Proof. Take $\underline{W}$ with $\mathsf{lev}(\underline{W}) \leq n$. Take $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\texttt{toEmul}(\Gamma)_{m;p}]\!]$. Then we need to prove that

$$(\underline{W}, \texttt{case}_{\to;m} \ (\text{upgrade}_{m;1} \ \mathbf{t_1}\gamma) \ \mathbf{t_2}\gamma, t_1\gamma \ t_2\gamma) \in \mathcal{E}[\![\texttt{EmulDV}_{m;p}]\!]_\square.$$

By Theorem 9, it follows from $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_1} \ \square_{\mathsf{n}} \ \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$ that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{upgrade}_{\mathsf{m;1}} \ \mathbf{t_1} \ \square_{\mathsf{n}} \ \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m+1;p}}$.

This gives us that

$$(\underline{\mathsf{W}}, \mathrm{upgrade}_{\mathsf{m;1}} \ \mathbf{t_1}\gamma, \mathsf{t}_1\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}.$$

By Lemma 19, it suffices to prove that for all $\underline{\mathsf{W}}' \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}}', \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$, that then

$$(\underline{\mathsf{W}}', \mathtt{case}_{\rightarrow;\mathtt{m}} \ \mathbf{v_1} \ \mathbf{t_2}\gamma, \mathsf{v}_1 \ \mathsf{t}_2\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

From $(\underline{\mathsf{W}}', \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$, we get by definition that one of the following cases must hold:

- $\mathbf{v_1} = \mathbf{in}_{\mathrm{unk};\mathbf{n}} \wedge p = \mathtt{imprecise}$
- $\exists \mathbf{v_1'}. \ \mathbf{v_1} = \mathbf{in}_{\mathcal{B};\mathbf{n}}(v_1') \wedge (\underline{\mathsf{W}}', \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$
- $\exists \mathbf{v_1'}. \ \mathbf{v_1} = \mathbf{in}_{\times;\mathbf{n}}(\mathbf{v_1'}) \wedge (\underline{\mathsf{W}}', \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \times \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$
- $\exists \mathbf{v_1'}. \ \mathbf{v_1} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathbf{v_1'}) \wedge (\underline{\mathsf{W}}', \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \uplus \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$
- $\exists \mathbf{v_1'}. \ \mathbf{v_1} = \mathbf{in}_{\rightarrow;\mathbf{n}}(\mathbf{v_1'}) \wedge (\underline{\mathsf{W}}', \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \rightarrow \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$

In the first case, we know that $\square = \lesssim$ and $\mathbb{E}[\mathtt{case}_{\rightarrow;\mathtt{m}} \ \mathbf{v_1} \ \mathbf{t_2}\gamma]\Uparrow$ for any $\mathbb{E}$. By definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ and by definition of $\mathsf{O}(\underline{\mathsf{W}}')_{\lesssim}$, the result follows.

In the second, third and fourth case, we also have that $\mathbb{E}[\mathtt{case}_{\rightarrow;\mathtt{m}} \ \mathbf{v_1} \ \mathbf{t_2}\gamma]\Uparrow$ for any $\mathbb{E}$. Additionally, we have that $\mathbb{E}[\mathsf{v}_1 \ \mathsf{t}_2\gamma] \hookrightarrow^* \mathsf{wrong}$ for any $\mathbb{E}$. The result follows by definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ and by Lemma 6.

In the fifth case, we have that $\mathbb{E}[\mathtt{case}_{\rightarrow;\mathtt{m}} \ \mathbf{v_1} \ \mathbf{t_2}\gamma] \hookrightarrow^* \mathbb{E}[\mathbf{v_1'} \ \mathbf{t_2}\gamma]$, so by Lemma 8, it suffices to prove that

$$(\underline{\mathsf{W}}', \mathbf{v_1'} \ \mathbf{t_2}\gamma, \mathsf{v}_1 \ \mathsf{t}_2\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

From $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_2} \ \square_{\mathsf{n}} \ \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}$, we have that

$$(\underline{\mathsf{W}}', \mathbf{t_2}\gamma, \mathsf{t}_2\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

By Lemma 19, it suffices to prove that for all $\underline{\mathsf{W}}'' \sqsupseteq \underline{\mathsf{W}}'$, $(\underline{\mathsf{W}}'', \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$, that then

$$(\underline{\mathsf{W}}'', \mathbf{v_1'} \ \mathbf{v_2}, \mathsf{v}_1 \ \mathsf{v}_2) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

By Lemma 13, we have that $(\underline{\mathsf{W}}'', \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{n;p}} \rightarrow \mathtt{EmulDV}_{\mathsf{n;p}}]\!]_{\square}$ and the result follows by Lemma 20. $\square$

Lemma 43 (Compatibility lemma of emulation for case)**.** *If ($m > n$ and $p =$ $\mathtt{precise}$) or ($\square = \lesssim$ and $p = \mathtt{imprecise}$), then we have that if* $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_1} \ \square_{\mathsf{n}} \ \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$, $\mathtt{toEmul}(\Gamma, \mathsf{x})_{\mathsf{m;p}} \vdash \mathbf{t_2} \ \square_{\mathsf{n}} \ \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}$, *and if* $\mathtt{toEmul}(\Gamma, \mathsf{x})_{\mathsf{m;p}} \vdash \mathbf{t_3} \ \square_{\mathsf{n}} \ \mathsf{t}_3 : \mathtt{EmulDV}_{\mathsf{m;p}}$, *then*

$$\begin{aligned} \mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \ & \mathrm{case} \ (\mathtt{case}_{\uplus;\mathtt{m}} \ (\mathrm{upgrade}_{\mathsf{n;1}} \ \mathbf{t_1})) \ \mathrm{of} \ \mathrm{inl} \ \mathbf{x} \mapsto \mathbf{t_2} \mid \mathrm{inr} \ \mathbf{x} \mapsto \mathbf{t_3} \ \square_{\mathsf{n}} \\ & \mathrm{case} \ \mathsf{t}_1 \ \mathrm{of} \ \mathrm{inl} \ \mathsf{x} \mapsto \mathsf{t}_2 \mid \mathrm{inr} \ \mathsf{x} \mapsto \mathsf{t}_3 : \mathtt{EmulDV}_{\mathsf{m;p}}. \end{aligned}$$

Proof. Take $\underline{W}$ with $\mathsf{lev}(\underline{W}) \leq n$. Take $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{m;p}]\!]$. Then we need to prove that

$$(\underline{W}, \text{case } (\mathtt{case}_{\uplus;m}\ (\text{upgrade}_{n;1}\ \mathbf{t_1}\gamma)) \text{ of inl } \mathbf{x} \mapsto \mathbf{t_2}\gamma \mid \text{inr } \mathbf{x} \mapsto \mathbf{t_3}\gamma,$$
$$\text{case } \mathsf{t}_1\gamma \text{ of inl } \mathsf{x} \mapsto \mathsf{t}_2\gamma \mid \text{inr } \mathsf{x} \mapsto \mathsf{t}_3\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{m;p}]\!]_\square.$$

By Theorem 9, it follows from $\mathtt{toEmul}(\Gamma)_{m;p} \vdash \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \mathtt{EmulDV}_{m;p}$ that $\mathtt{toEmul}(\Gamma)_{m;p} \vdash \text{upgrade}_{m;1}\ \mathbf{t_1} \ \square_n \ \mathsf{t}_1 : \mathtt{EmulDV}_{m+1;p}$.

This gives us that

$$(\underline{W}, \text{upgrade}_{m;1}\ \mathbf{t_1}\gamma, \mathsf{t}_1\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{m+1;p}]\!]_\square.$$

By Lemma 19, it suffices to prove that for all $\underline{W'} \sqsupseteq \underline{W}$, $(\underline{W'}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{m+1;p}]\!]_\square$, that then

$$(\underline{W'}, \text{case } (\mathtt{case}_{\uplus;m}\ \mathbf{v_1}) \text{ of inl } \mathbf{x} \mapsto \mathbf{t_2}\gamma \mid \text{inr } \mathbf{x} \mapsto \mathbf{t_3}\gamma,$$
$$\text{case } \mathsf{v}_1 \text{ of inl } \mathsf{x} \mapsto \mathsf{t}_2\gamma \mid \text{inr } \mathsf{x} \mapsto \mathsf{t}_3\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{m;p}]\!]_\square.$$

From $(\underline{W'}, \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{m+1;p}]\!]_\square$, we get by definition that one of the following cases must hold:

- $\mathbf{v_1} = \mathbf{in}_{\text{unk};\mathbf{n}} \wedge p = \mathtt{imprecise}$
- $\exists \mathbf{v_1'}.\ \mathbf{v_1} = \mathbf{in}_{\mathcal{B};\mathbf{n}}(v_1') \wedge (\underline{W'}, \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathcal{B}]\!]_\square$
- $\exists \mathbf{v_1'}.\ \mathbf{v_1} = \mathbf{in}_{\times;\mathbf{n}}(\mathbf{v_1'}) \wedge (\underline{W'}, \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{m;p} \times \mathtt{EmulDV}_{m;p}]\!]_\square$
- $\exists \mathbf{v_1'}.\ \mathbf{v_1} = \mathbf{in}_{\uplus;\mathbf{n}}(\mathbf{v_1'}) \wedge (\underline{W'}, \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{m;p} \uplus \mathtt{EmulDV}_{m;p}]\!]_\square$
- $\exists \mathbf{v_1'}.\ \mathbf{v_1} = \mathbf{in}_{\to;\mathbf{n}}(\mathbf{v_1'}) \wedge (\underline{W'}, \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{m;p} \to \mathtt{EmulDV}_{m;p}]\!]_\square$

In the first case, we know that $\square = \lesssim$ and $\mathbb{E}[\text{case } (\mathtt{case}_{\uplus;m}\ \mathbf{v_1}) \text{ of inl } \mathbf{x} \mapsto \mathbf{t_2}\gamma \mid \text{inr } \mathbf{x} \mapsto \mathbf{t_3}\gamma]\Uparrow$ for any $\mathbb{E}$. By definition of $\mathcal{E}[\![\mathtt{EmulDV}_{m;p}]\!]_\square$ and by definition of $\mathsf{O}(\underline{W'})_\lesssim$, the result follows.

In the second, third and fifth case, we also have that $\mathbb{E}[\text{case } (\mathtt{case}_{\uplus;m}\ \mathbf{v_1}) \text{ of inl } \mathbf{x} \mapsto \mathbf{t_2}\gamma \mid \text{inr } \mathbf{x} \mapsto \mathbf{t_3}\gamma]\Uparrow$ for any $\mathbb{E}$. Additionally, we have that $\mathbb{E}[\text{case } \mathsf{v}_1 \text{ of inl } \mathsf{x} \mapsto \mathsf{t}_2\gamma \mid \text{inr } \mathsf{x} \mapsto \mathsf{t}_3\gamma] \hookrightarrow^* \mathsf{wrong}$ for any $\mathbb{E}$. The result follows by definition of $\mathcal{E}[\![\mathtt{EmulDV}_{m;p}]\!]_\square$ and by Lemma 6.

In the fourth case, we get from $(\underline{W'}, \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{m;p} \uplus \mathtt{EmulDV}_{m;p}]\!]_\square$ values $\mathbf{v_1''}$ and v_1'' such that $(\underline{W}\ \mathbf{v_1''}, \mathsf{v}_1'') \in \triangleright \mathcal{V}[\![\mathtt{EmulDV}_{m;p}]\!]_\square$ and either $(\mathbf{v_1'} = \text{inl } \mathbf{v_1''}$ and $\mathsf{v}_1 = \text{inl } \mathsf{v}_1'')$ or $(\mathbf{v_1'} = \text{inr } \mathbf{v_1''}$ and $\mathsf{v}_1 = \text{inr } \mathsf{v}_1'')$. We only consider the first case further, the other case is completely similar.

We now have that

$$\mathbb{E}[\text{case } (\mathtt{case}_{\uplus;m}\ \mathbf{v_1}) \text{ of inl } \mathbf{x} \mapsto \mathbf{t_2}\gamma \mid \text{inr } \mathbf{x} \mapsto \mathbf{t_3}\gamma] \hookrightarrow$$
$$\mathbb{E}[\text{case } \mathbf{v_1'} \text{ of inl } \mathbf{x} \mapsto \mathbf{t_2}\gamma \mid \text{inr } \mathbf{x} \mapsto \mathbf{t_3}\gamma] \hookrightarrow \mathbb{E}[\mathbf{t_2}\gamma[\mathbf{v_1''}/\mathbf{x}]]$$

and

$$\mathbb{E}[\text{case } \mathsf{v}_1 \text{ of inl } \mathsf{x} \mapsto \mathsf{t}_2\gamma \mid \text{inr } \mathsf{x} \mapsto \mathsf{t}_3\gamma] \hookrightarrow \mathsf{t}_2\gamma[\mathsf{v}_1''/\mathsf{x}].$$

Now if $\mathsf{lev}(\underline{W}') = 0$, then we have that

$$(\underline{W}', \text{case } (\mathbf{case}_{\uplus;m}\ \mathbf{v_1}) \text{ of inl } \mathbf{x} \mapsto \mathbf{t_2}\gamma \mid \text{inr } \mathbf{x} \mapsto \mathbf{t_3}\gamma,$$
$$\text{case } \mathsf{v_1} \text{ of inl } \mathsf{x} \mapsto \mathsf{t_2}\gamma \mid \text{inr } \mathsf{x} \mapsto \mathsf{t_3}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square,$$

by definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$ and Lemma 7.

If $\mathsf{lev}(\underline{W}') > 0$, then we have that $(\triangleright \underline{W}', \mathbf{v''_1}, \mathsf{v''_1}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$. By Lemma 8, it suffices to prove that

$$(\triangleright \underline{W}', \mathbf{t_2}\gamma[\mathbf{v''_1}/\mathbf{x}], \mathsf{t_2}\gamma[\mathsf{v''_1}/\mathsf{x}]) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square.$$

This follows from $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_1} \ \square_\mathsf{n}\ \mathsf{t_1} : \mathtt{EmulDV}_{\mathsf{m;p}}$ since $\mathsf{lev}(\triangleright \underline{W}') \leq \mathsf{lev}(\underline{W}') \leq \mathsf{lev}(\underline{W}) \leq n$ if we show that $(\triangleright \underline{W}', \gamma[\mathbf{x} \mapsto \mathbf{v''_1}], \gamma[\mathsf{x} \mapsto \mathsf{v''_1}]) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$.

We know that $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]$, and by Lemma 11, also $(\triangleright \underline{W}', \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]$. Combined with $(\triangleright \underline{W}', \mathbf{v''_1}, \mathsf{v''_1}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$, this gives us $(\triangleright \underline{W}', \gamma[\mathbf{x} \mapsto \mathbf{v''_1}], \gamma[\mathsf{x} \mapsto \mathsf{v''_1}]) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$, as required. ☐

Lemma 44 (Compatibility lemma of emulation for pair)**.** *If ($m > n$ and $p =$* `precise`*) or ($\square = \lesssim$ and $p =$* `imprecise`*), then we have that if* $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_1} \ \square_\mathsf{n}\ \mathsf{t_1} : \mathtt{EmulDV}_{\mathsf{m;p}}$ *and* $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_2} \ \square_\mathsf{n}\ \mathsf{t_2} : \mathtt{EmulDV}_{\mathsf{m;p}}$*, then*

$$\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \text{downgrade}_{\mathsf{m;1}}\ (\mathbf{in}_{\times;\mathbf{m}}\ \langle \mathbf{t_1}, \mathbf{t_2}\rangle) \ \square_\mathsf{n}\ \langle \mathsf{t_1}, \mathsf{t_2}\rangle : \mathtt{EmulDV}_{\mathsf{m;p}}.$$

Proof. By Theorem 9, it suffices to prove that

$$\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash (\mathbf{in}_{\times;\mathbf{m}}\ \langle \mathbf{t_1}, \mathbf{t_2}\rangle) \ \square_\mathsf{n}\ \langle \mathsf{t_1}, \mathsf{t_2}\rangle : \mathtt{EmulDV}_{\mathsf{m+1;p}}.$$

Take $\underline{W}$ such that $\mathsf{lev}(\underline{W}) \leq n$ and $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$. Then we need to show that

$$(\underline{W}, \mathbf{in}_{\times;\mathbf{m}}\ \langle \mathbf{t_1}\gamma, \mathbf{t_2}\gamma\rangle, \langle \mathsf{t_1}\gamma, \mathsf{t_2}\gamma\rangle) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_\square.$$

From $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_1} \ \square_\mathsf{n}\ \mathsf{t_1} : \mathtt{EmulDV}_{\mathsf{m;p}}$, $\mathsf{lev}(\underline{W}) \leq n$ and $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$, we get that

$$(\underline{W}, \mathbf{t_1}\gamma, \mathsf{t_1}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square.$$

By Lemma 19, it then suffices to prove that for all $\underline{W}' \sqsupseteq \underline{W}$, $(\underline{W}', \mathbf{v_1}, \mathsf{v_1}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$, we have that

$$(\underline{W}', \mathbf{in}_{\times;\mathbf{m}}\ \langle \mathbf{v_1}, \mathbf{t_2}\gamma\rangle, \langle \mathsf{v_1}, \mathsf{t_2}\gamma\rangle) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_\square.$$

By Lemma 11, we have that $(\underline{W}', \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$ from $\underline{W}' \sqsupseteq \underline{W}$. From this, from $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_2} \ \square_\mathsf{n}\ \mathsf{t_2} : \mathtt{EmulDV}_{\mathsf{m;p}}$ and $\mathsf{lev}(\underline{W}') \leq \mathsf{lev}(\underline{W}) \leq n$, we then get

$$(\underline{W}', \mathbf{t_2}\gamma, \mathsf{t_2}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square.$$

By Lemma 19, it then suffices to prove that for all $\underline{W}'' \sqsupseteq \underline{W}'$, $(\underline{W}'', \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$, we have that

$$(\underline{W}'', \mathbf{in}_{\times;\mathbf{m}}\ \langle \mathbf{v_1}, \mathbf{v_2} \rangle, \langle \mathsf{v}_1, \mathsf{v}_2 \rangle) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square},$$

or (by Lemma 10)

$$(\underline{W}'', \mathbf{in}_{\times;\mathbf{m}}\ \langle \mathbf{v_1}, \mathbf{v_2} \rangle, \langle \mathsf{v}_1, \mathsf{v}_2 \rangle) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}.$$

By definition of $\mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$, it suffices to prove that $\langle \mathbf{v_1}, \mathbf{v_2} \rangle$ is $\mathbf{oftype}(\mathtt{EmulDV}_{\mathsf{m;p}} \times \mathtt{EmulDV}_{\mathsf{m;p}})$, which follows from the hypotheses on $\mathbf{v_1}$ and $\mathbf{v_2}$ and by rule λ^{τ}-Type-pair, and

$$(\underline{W}'', \langle \mathbf{v_1}, \mathbf{v_2} \rangle, \langle \mathsf{v}_1, \mathsf{v}_2 \rangle) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}} \times \mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

This follows by definition, by Lemma 13, and by the facts that $(\underline{W}', \mathbf{v_1}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ and $(\underline{W}'', \mathbf{v_2}, \mathsf{v}_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$. ∎

Lemma 45 (Compatibility lemma of emulation for injection)**.** *If ($m > n$ and $p =$ `precise`) or ($\square = \lesssim$ and $p =$ `imprecise`), then we have that if* $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t} \ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$*, then*

$$\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{downgrade}_{\mathsf{m;1}}\ (\mathbf{in}_{\uplus;\mathbf{m}}\ (\mathrm{inl}\ \mathbf{t}))\ \square_{\mathsf{n}}\ \mathrm{inl}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}.$$

and

$$\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{downgrade}_{\mathsf{m;1}}\ (\mathbf{in}_{\uplus;\mathbf{m}}\ (\mathrm{inr}\ \mathbf{t}))\ \square_{\mathsf{n}}\ \mathrm{inr}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}.$$

Proof. We only prove the result about inr , the other is completely similar.

By Theorem 9, it suffices to prove that

$$\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{in}_{\uplus;\mathbf{m}}\ (\mathrm{inl}\ \mathbf{t})\ \square_{\mathsf{n}}\ \mathrm{inl}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m+1;p}}.$$

Take $\underline{W}$ such that $\mathsf{lev}(\underline{W}) \leq n$ and $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_{\square}$. Then we need to show that

$$(\underline{W}, \mathbf{in}_{\uplus;\mathbf{m}}\ (\mathrm{inl}\ \mathbf{t}\gamma), \mathrm{inl}\ \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}.$$

From $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$, $\mathsf{lev}(\underline{W}) \leq n$ and $(\underline{W}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_{\square}$, we get that

$$(\underline{W}, \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

By Lemma 19, it then suffices to prove that for all $\underline{W}' \sqsupseteq \underline{W}$, $(\underline{W}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$, we have that

$$(\underline{W}', \mathbf{in}_{\times;\mathbf{m}}\ (\mathrm{inl}\ \mathbf{v}), \mathrm{inl}\ \mathsf{v}) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square},$$

or, by Lemma 10,

$$(\underline{W}', \mathbf{in}_{\times;\mathbf{m}}\ (\mathrm{inl}\ \mathbf{v}), \mathrm{inl}\ \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}.$$

By definition of $\mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$, it suffices to prove that $\mathrm{inl}\ \mathbf{v}$ is $\mathbf{oftype}()$, which follows from the hypothesis on $\mathbf{v}$ and rule λ^{τ}-Type-inl, and

$$(\underline{W}', \mathrm{inl}\ \mathbf{v}, \mathrm{inl}\ \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}} \uplus \mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

This follows by definition and by the fact that $(\underline{W}', \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$. ∎

Lemma 46 (Compatibility lemma of emulation for projection). *If* ($m > n$ *and* $p = \texttt{precise}$) *or* ($\square = \lesssim$ *and* $p = \texttt{imprecise}$), *then we have that if* $\texttt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t} \ \square_{\mathsf{n}} \ \mathsf{t} : \texttt{EmulDV}_{\mathsf{m;p}}$, *then*

$$\texttt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash (\texttt{case}_{\times;\texttt{m}} \ (\text{upgrade}_{\mathsf{m;1}} \ \mathbf{t})).\mathbf{1} \ \square_{\mathsf{n}} \ \mathsf{t.1} : \texttt{EmulDV}_{\mathsf{m;p}}.$$

and

$$\texttt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash (\texttt{case}_{\times;\texttt{m}} \ (\text{upgrade}_{\mathsf{m;1}} \ \mathbf{t})).\mathbf{2} \ \square_{\mathsf{n}} \ \mathsf{t.2} : \texttt{EmulDV}_{\mathsf{m;p}}.$$

Proof. We only prove the result about $\mathbf{t.1}$ and $\mathsf{t.1}$, the other is completely similar.

Take $\underline{\mathsf{W}}$ such that $\mathsf{lev}(\underline{\mathsf{W}}) \leq n$ and $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\texttt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_{\square}$. Then we need to show that

$$(\underline{\mathsf{W}}, (\texttt{case}_{\times;\texttt{m}} \ (\text{upgrade}_{\mathsf{m;1}} \ \mathbf{t}\gamma)).\mathbf{1}, (\mathsf{t}\gamma).1) \in \mathcal{E}[\![\texttt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}.$$

From $\texttt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t} \ \square_{\mathsf{n}} \ \mathsf{t} : \texttt{EmulDV}_{\mathsf{m;p}}$, we get by Theorem 9 that $\texttt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \text{upgrade}_{\mathsf{m;1}} \ \mathbf{t} \ \square_{\mathsf{n}} \ \mathsf{t} : \texttt{EmulDV}_{\mathsf{m+1;p}}$. From $\mathsf{lev}(\underline{\mathsf{W}}) \leq n$ and $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\texttt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_{\square}$, we then get that

$$(\underline{\mathsf{W}}, \text{upgrade}_{\mathsf{m;1}} \ \mathbf{t}\gamma, \mathsf{t}\gamma) \in \mathcal{E}[\![\texttt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}.$$

By Lemma 19, it then suffices to prove that for all $\underline{\mathsf{W}'} \sqsupseteq \underline{\mathsf{W}}$, $(\underline{\mathsf{W}'}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$, we have that

$$(\underline{\mathsf{W}'}, (\texttt{case}_{\times;\texttt{m}} \ \mathbf{v}).\mathbf{1}, \mathsf{v}.1) \in \mathcal{E}[\![\texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

From $(\underline{\mathsf{W}'}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$, we get that one of the following cases must hold:

- $\mathbf{v} = \mathbf{in}_{\mathbf{unk;m}} \wedge p = \texttt{imprecise}$
- $\exists \mathbf{v}'. \ \mathbf{v} = \mathbf{in}_{\mathcal{B};\mathbf{m}}(v') \wedge (\underline{\mathsf{W}'}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathcal{B}]\!]_{\square}$
- $\exists \mathbf{v}'. \ \mathbf{v} = \mathbf{in}_{\times;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}'}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{m;p}} \times \texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$
- $\exists \mathbf{v}'. \ \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}'}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{m;p}} \uplus \texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$
- $\exists \mathbf{v}'. \ \mathbf{v} = \mathbf{in}_{\rightarrow;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}'}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{m;p}} \rightarrow \texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$

In the first case, we have that $\mathbb{E}[(\texttt{case}_{\times;\texttt{m}} \ \mathbf{v}).\mathbf{1}]\Uparrow$ for any $\mathbb{E}$. We then also know that $\square = \lesssim$, and by definition of $\mathcal{E}[\![\texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ and $\mathsf{O}(\underline{\mathsf{W}'})_{\lesssim}$, the result follows.

In the second, fourth and fifth case, we have that $\mathbb{E}[(\texttt{case}_{\times;\texttt{m}} \ \mathbf{v}).\mathbf{1}]\Uparrow$ for any $\mathbb{E}$ and $\mathbb{E}[\mathsf{v}.1] \hookrightarrow^{*} \mathsf{wrong}$ for any $\mathbb{E}$. By the definition of $\mathcal{E}[\![\texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ and Lemma 6, the result follows.

In the third case, from $(\underline{\mathsf{W}'}, \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{m;p}} \times \texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$, we get $\mathbf{v}'_{\mathbf{1}}, \mathbf{v}'_{\mathbf{2}}, \mathsf{v}_1, \mathsf{v}_2$ such that $\mathbf{v}' = \langle \mathbf{v}'_{\mathbf{1}}, \mathbf{v}'_{\mathbf{2}} \rangle$ and $\mathsf{v} = \langle \mathsf{v}_1, \mathsf{v}_2 \rangle$, $(\underline{\mathsf{W}'}, \mathbf{v}'_{\mathbf{1}}, \mathsf{v}_1) \in \triangleright \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ and $(\underline{\mathsf{W}'}, \mathbf{v}'_{\mathbf{2}}, \mathsf{v}_2) \in \triangleright \mathcal{V}[\![\texttt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$.

We then have that

$$\mathbb{E}[(\texttt{case}_{\times;\texttt{m}} \ \mathbf{v}).\mathbf{1}] \hookrightarrow \mathbb{E}[\mathbf{v}'.\mathbf{1}] \hookrightarrow \mathbb{E}[\mathbf{v}'_{\mathbf{1}}]$$

for any $\mathbb{E}$ and

$$\mathbb{E}[\mathsf{v}.1] \hookrightarrow \mathbb{E}[\mathsf{v}_1]$$

for any $\mathbb{E}$.

Now if $\mathsf{lev}(\underline{\mathsf{W}}') = 0$, then we have that

$$(\underline{\mathsf{W}}', (\mathtt{case}_{\times;m}\ \mathbf{v}).\mathbf{1}, \mathsf{v}.1) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$$

by definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ and Lemma 7.

If $\mathsf{lev}(\underline{\mathsf{W}}') > 0$, then we have that $(\triangleright \underline{\mathsf{W}}', \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$ and $(\triangleright \underline{\mathsf{W}}', \mathbf{v_2'}, \mathsf{v}_2) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$. By Lemma 8, it suffices to prove that

$$(\triangleright \underline{\mathsf{W}}', \mathbf{v_1'}, \mathsf{v}_1) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}.$$

This follows directly using Lemma 10. $\square$

Lemma 47 (Compatibility lemma of emulation for if). *If ($m > n$ and $p =$ `precise`) or ($\square = \lesssim$ and $p =$ `imprecise`), then we have that if $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$ (H) and $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_1}\ \square_{\mathsf{n}}\ \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$ (H1) and $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_2}\ \square_{\mathsf{n}}\ \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}$ (H2), then*

$$\begin{aligned}\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{if}\ (\mathtt{case}_{\mathtt{Bool};n}(\mathrm{upgrade}_{\mathsf{n};1}(\mathbf{t})))\ \mathrm{then}\ \mathbf{t_1}\ \mathrm{else}\ \mathbf{t_2}\ \square_{\mathsf{n}} \\ \mathrm{if}\ \mathsf{t}\ \mathrm{then}\ \mathsf{t}_1\ \mathrm{else}\ \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}.\end{aligned}$$

Proof. Take $\underline{\mathsf{W}}$, $\mathsf{lev}(\underline{\mathsf{W}}) \le n$ (HN) and $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_{\square}$ (HG). We need to show that $(\underline{\mathsf{W}}, \mathrm{if}\ (\mathtt{case}_{\mathtt{Bool};n}(\mathrm{upgrade}_{\mathsf{n};1}(\mathbf{t})))\ \mathrm{then}\ \mathbf{t_1}\ \mathrm{else}\ \mathbf{t_2}, \mathrm{if}\ \mathsf{t}\ \mathrm{then}\ \mathsf{t}_1\ \mathrm{else}\ \mathsf{t}_2) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$.

Apply Theorem 9 to H to get that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{upgrade}_{\mathsf{n};1}\mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m+1;p}}$ (HH). By HH, HN and HG, we have that $(\underline{\mathsf{W}}, \mathrm{upgrade}_{\mathsf{n};1}(\mathbf{t}\gamma), \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$.

Assume $\mathrm{A} = \forall \underline{\mathsf{W}}_f \sqsupseteq \underline{\mathsf{W}}, \forall (\underline{\mathsf{W}}_f, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$ (HV), $(\mathbb{E}[\mathrm{if}\ \mathtt{case}_{\mathtt{Bool};n}\ \cdot\ \mathrm{then}\ \mathbf{t_1}\gamma\ \mathrm{else}\ \mathbf{t_2}\gamma], \mathbb{E}[\mathrm{if}\ \cdot\ \mathrm{then}\ \mathsf{t}_1\gamma\ \mathrm{else}\ \mathsf{t}_2\gamma]) \in \mathcal{K}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_{\square}$.

The thesis follows from Lemma 8.

Prove A. Let $\mathbb{E}'\cdot = \mathbb{E}[\mathrm{if}\ \mathtt{case}_{\mathtt{Bool};n}\ \cdot\ \mathrm{then}\ \mathbf{t_1}\gamma\ \mathrm{else}\ \mathbf{t_2}\gamma]$ and $\mathbb{E}'\cdot = \mathbb{E}[\mathrm{if}\ \cdot\ \mathrm{then}\ \mathsf{t}_1\gamma\ \mathrm{else}\ \mathsf{t}_2\gamma]) \in \mathcal{K}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]$. We have these cases based on HV:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};\mathbf{m}} \wedge p = \mathtt{imprecise}$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\mathtt{Unit};\mathbf{m}}(v') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{Unit}]\!]_{\square}$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\mathtt{Bool};\mathbf{m}}(v') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{Bool}]\!]_{\square}$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\times;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}} \times \mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}} \uplus \mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$
- $\exists \mathbf{v}'.\ \mathbf{v} = \mathbf{in}_{\to;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}} \to \mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$

In the first case, we have that $\mathbb{E}'[\mathbf{v}]\Uparrow$ for any $\mathbb{E}$. We then also know that $\square = \lesssim$, and by definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$ and $\mathsf{O}(\underline{\mathsf{W}}')_\lesssim$, the result follows.

In the seconfd, fourth, fifth and sixth case, we have that $\mathbb{E}'[\mathbf{v}]\Uparrow$ for any $\mathbb{E}$ and $\mathbb{E}'[\mathsf{v}] \hookrightarrow^* \mathsf{wrong}$ for any $\mathbb{E}$. By the definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$ and Lemma 6, the result follows.

In the third case we have two cases: $\mathbf{v}' \equiv \mathsf{v}' \equiv \mathtt{true}$ or $\mathbf{v}' \equiv \mathsf{v}' \equiv \mathtt{false}$. We consider the first only, the second is dual with H2 used in place of H1.

We have that $\mathbb{E}'[\mathbf{in}_{\mathtt{Bool};\mathbf{m}}(\mathbf{v}')] \hookrightarrow^* \mathbb{E}[\mathbf{t_1}\gamma]$ and $\mathbb{E}'[\mathsf{v}] \hookrightarrow \mathbb{E}[\mathsf{t}_1\gamma]$. Assume B$=$ $(\mathbb{E}[\mathbf{t_1}\gamma], \mathbb{E}[\mathsf{t}_1\gamma]) \in \mathsf{O}(\triangleright\,\underline{\mathsf{W}}_f)$, the thesis follows from Lemma 8.

Prove B. Unfold H1 and we get $\forall \underline{\mathsf{W}}_1$, $\forall(\underline{\mathsf{W}}_1, \gamma_1, \gamma_1) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$, $\forall(\underline{\mathsf{W}}_1, \mathbb{E}_1, \mathbb{E}_1) \in \mathcal{K}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]$ (HJ), $(\mathbb{E}_1[\mathbf{t_1}\gamma_1], \mathbb{E}_1[\mathsf{t}_1\gamma_1]) \in \mathsf{O}(\triangleright\,\underline{\mathsf{W}}_1)$.

The thesis holds by instantiating $\underline{\mathsf{W}}_1$ with $\triangleright\,\underline{\mathsf{W}}_f$, γ_1 with γ, γ_1 with γ, $\mathbb{E}_1$ with $\mathbb{E}$ and $\mathbb{E}_1$ with $\mathbb{E}$ and by Lemma 12 applied to HJ. ◻

Lemma 48 (Compatibility lemma of emulation for sequence)**.** *If ($m > n$ and $p = \mathtt{precise}$) or ($\square = \lesssim$ and $p = \mathtt{imprecise}$), then we have that if* $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t} \;\square_{\mathsf{n}}\; \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$ *and* $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{t_1} \;\square_{\mathsf{n}}\; \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$, *then*

$$\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}(\mathtt{case}_{\mathtt{Unit};\mathtt{n}}\ (\mathrm{upgrade}_{\mathsf{n};1}(\mathbf{t}))); \mathbf{t_1} \;\square_{\mathsf{n}}\; \mathsf{t}; \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}.$$

Proof. Take $\underline{\mathsf{W}}$, $\mathsf{lev}(\underline{\mathsf{W}}) \leq n$ (HN) and $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$ (HG). We need to show that $(\underline{\mathsf{W}}, (\mathtt{case}_{\mathtt{Unit};\mathtt{n}}(\mathrm{upgrade}_{\mathsf{n};1}(\mathbf{t}))); \mathbf{t_1}, \mathsf{t}; \mathsf{t}_1) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$.

Apply Theorem 9 to H to get that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{upgrade}_{\mathsf{n};1}\mathbf{t} \;\square_{\mathsf{n}}\; \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m+1;p}}$ (HH). By HH, HN and HG, we have that $(\underline{\mathsf{W}}, \mathrm{upgrade}_{\mathsf{n};1}(\mathbf{t}\gamma), \mathsf{t}\gamma) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_\square$.

Assume A $= \forall \underline{\mathsf{W}}_f \sqsupseteq \underline{\mathsf{W}}, \forall(\underline{\mathsf{W}}_f, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_\square$ (HV), $(\mathbb{E}[\mathtt{case}_{\mathtt{Unit};\mathtt{n}}\cdot; \mathbf{t_1}\gamma]$, $\mathbb{E}[\cdot; \mathsf{t}_1\gamma]) \in \mathcal{K}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_\square$.

The thesis follows from Lemma 8.

Prove A. Let $\mathbb{E}'\cdot = \mathbb{E}[\mathtt{case}_{\mathtt{Unit};\mathtt{n}}\cdot; \mathbf{t_1}\gamma]$ and $\mathbb{E}'\cdot = \mathbb{E}[\cdot; \mathsf{t}_1\gamma]) \in \mathcal{K}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]$. We have these cases based on HV:

- $\mathbf{v} = \mathbf{in}_{\mathrm{unk};\mathbf{m}} \wedge p = \mathtt{imprecise}$
- $\exists \mathbf{v}'. \mathbf{v} = \mathbf{in}_{\mathtt{Unit};\mathbf{m}}(v') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{Unit}]\!]_\square$
- $\exists \mathbf{v}'. \mathbf{v} = \mathbf{in}_{\mathtt{Bool};\mathbf{m}}(v') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{Bool}]\!]_\square$
- $\exists \mathbf{v}'. \mathbf{v} = \mathbf{in}_{\times;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}} \times \mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$
- $\exists \mathbf{v}'. \mathbf{v} = \mathbf{in}_{\uplus;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}} \uplus \mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$
- $\exists \mathbf{v}'. \mathbf{v} = \mathbf{in}_{\rightarrow;\mathbf{m}}(\mathbf{v}') \wedge (\underline{\mathsf{W}}', \mathbf{v}', \mathsf{v}) \in \mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m;p}} \rightarrow \mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$

In the first case, we have that $\mathbb{E}'[\mathbf{v}]\Uparrow$ for any $\mathbb{E}$. We then also know that $\square = \lesssim$, and by definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$ and $\mathsf{O}(\underline{\mathsf{W}}')_\lesssim$, the result follows.

In the third, fourth, fifth and sixth case, we have that $\mathbb{E}'[\mathbf{v}]\Uparrow$ for any $\mathbb{E}$ and $\mathbb{E}'[\mathsf{v}] \hookrightarrow^* \mathsf{wrong}$ for any $\mathbb{E}$. By the definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$ and Lemma 6, the result follows.

In the second case we have that: $\mathbf{v}' \equiv \mathsf{v} \equiv \mathtt{unit}$.

We have that $\mathbb{E}'[\mathbf{in}_{\mathtt{Unit};\mathbf{m}}(\mathbf{v}')] \hookrightarrow^* \mathbb{E}[\mathbf{t_1}\gamma]$ and $\mathbb{E}'[\mathsf{v}] \hookrightarrow \mathbb{E}[\mathsf{t}_1\gamma]$. Assume B$=$ $(\mathbb{E}[\mathbf{t_1}\gamma], \mathbb{E}[\mathsf{t}_1\gamma]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}}_f)$, the thesis follows from Lemma 8.

Prove B. Unfold H1 and we get $\forall \underline{\mathsf{W}}_1$, $\forall (\underline{\mathsf{W}}_1, \boldsymbol{\gamma_1}, \gamma_1) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$, $\forall (\underline{\mathsf{W}}_1, \mathbb{E}_\mathbf{1}, \mathbb{E}_1) \in \mathcal{K}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$ (HJ), $(\mathbb{E}_\mathbf{1}[\mathbf{t_1}\boldsymbol{\gamma_1}], \mathbb{E}_1[\mathsf{t}_1\gamma_1]) \in \mathsf{O}(\triangleright \underline{\mathsf{W}}_1)$.

The thesis holds by instantiating $\underline{\mathsf{W}}_1$ with $\triangleright \underline{\mathsf{W}}_f$, $\boldsymbol{\gamma_1}$ with γ, γ_1 with γ, $\mathbb{E}_\mathbf{1}$ with $\mathbb{E}$ and $\mathbb{E}_1$ with $\mathbb{E}$ and by Lemma 12 applied to HJ. ☐

Theorem 11 (Emulate is semantics-preserving)**.** *If $\Gamma \vdash \mathsf{t}$, and if ($m > n$ and $p = \mathtt{precise}$) or ($\square =\lesssim$ and $p = \mathtt{imprecise}$), then we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_\mathsf{m}(\mathsf{t}) \ \square_\mathsf{n}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$.*

Proof. By induction on $\Gamma \vdash \mathsf{t}$.

- rule λ^u-Wf-Base: We have that

  $$\mathrm{emulate}_\mathsf{m}(\mathsf{b}) \stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{m};1}\ (\mathbf{in}_{\mathcal{B},\mathbf{m}}\ \mathtt{b})$$

  By Theorem 9, it suffices to prove that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathbf{in}_{\mathcal{B};\mathbf{m}}\ \mathtt{b}\ \square_\mathsf{n}\ \mathsf{b} : \mathtt{EmulDV}_{\mathsf{m+1;p}}$.

  So, take $\underline{\mathsf{W}}$ with $\mathsf{lev}(\underline{\mathsf{W}}) \leq n$, $(\underline{\mathsf{W}}, \boldsymbol{\gamma}, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$. we need to show that $(\underline{\mathsf{W}}, \mathbf{in}_{\mathcal{B};\mathbf{m}}(\mathtt{b}), \mathsf{b}) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_\square$. This follows by the definition of $\mathcal{V}[\![\mathtt{EmulDV}_{\mathsf{m+1;p}}]\!]_\square$ and $\mathcal{V}[\![\mathcal{B}]\!]_\square$.

- rule λ^u-Wf-Lam: We have that

  $$\mathrm{emulate}_\mathsf{m}(\lambda \mathsf{x}.\, \mathsf{t}) \stackrel{\mathsf{def}}{=} \mathrm{downgrade}_{\mathsf{m};1}\ (\mathbf{in}_{\to;\mathbf{m}}\ (\lambda \mathbf{x} : \mathrm{UVal}_\mathsf{m}.\, \mathrm{emulate}_{\Gamma,\mathsf{x};\mathsf{m}}(\mathsf{t})))$$

  We get by induction that $\mathtt{toEmul}(\Gamma, \mathsf{x}])_{\mathsf{m;p}} \vdash \mathrm{emulate}_\mathsf{m}(\mathsf{t})\ \square_\mathsf{n}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$. The result follows by Lemma 41.

- rule λ^u-Wf-Var: We have that $\mathrm{emulate}_\mathsf{m}(\mathsf{x}) = \mathbf{x}$. So, take $\underline{\mathsf{W}}$ with $\mathsf{lev}(\underline{\mathsf{W}}) \leq n$ and $(\underline{\mathsf{W}}, \boldsymbol{\gamma}, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$. Then we need to show that $(\underline{\mathsf{W}}, \boldsymbol{\gamma}(\mathbf{x}), \gamma(\mathsf{x})) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_\square$. But since $\mathsf{x} \in \Gamma$, this follows directly from Lemma 10 and the definition of $\mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_\square$.

- rule λ^u-Wf-Pair: We have that

  $$\mathrm{emulate}_\mathsf{m}(\langle \mathsf{t}_1, \mathsf{t}_2 \rangle) = \mathrm{downgrade}_{\mathsf{m};1}\ (\mathbf{in}_{\times;\mathbf{m}}\ \langle \mathrm{emulate}_\mathsf{m}(\mathsf{t}_1), \mathrm{emulate}_\mathsf{m}(\mathsf{t}_2) \rangle).$$

  By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_\mathsf{m}(\mathsf{t}_1)\ \square_\mathsf{n}\ \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$ and $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_\mathsf{m}(\mathsf{t}_1)\ \square_\mathsf{n}\ \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}$. The result follows by Lemma 44.

- rule λ^u-Wf-Inl: We have that

  $$\mathrm{emulate}_\mathsf{m}(\mathrm{inl}\ \mathsf{t}) = \mathrm{downgrade}_{\mathsf{m};1}\ (\mathbf{in}_{\uplus;\mathbf{m}}\ (\mathrm{inl}\ (\mathrm{emulate}_\mathsf{m}(\mathbf{t_1})))).$$

  By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_\mathsf{m}(\mathsf{t})\ \square_\mathsf{n}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$ The result follows by Lemma 45.

- rule λ^{u}-Wf-Inr: We have that

$$\mathrm{emulate}_{\mathsf{m}}(\mathrm{inl}\ \mathsf{t}) = \mathrm{downgrade}_{\mathsf{m};1}\ (\mathbf{in}_{\uplus;\mathbf{m}}\ (\mathrm{inl}\ (\mathrm{emulate}_{\mathsf{m}}(\mathbf{t_1})))).$$

By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t})\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$ The result follows by Lemma 45.

- rule λ^{u}-Wf-App: We have that

$$\mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_1\ \mathsf{t}_2) \stackrel{\mathsf{def}}{=} \mathtt{case}_{\rightarrow;\mathtt{m}}\ (\mathrm{upgrade}_{\mathsf{m};1}\ \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_1))\ \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_2).$$

By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_1)\ \square_{\mathsf{n}}\ \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$, and $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_2)\ \square_{\mathsf{n}}\ \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}$. By Lemma 42, the result follows.

- rule λ^{u}-Wf-Proj1: We have that

$$\mathrm{emulate}_{\mathsf{m}}(\mathsf{t}.1) = (\mathtt{case}_{\times;\mathtt{m}}\ (\mathrm{upgrade}_{\mathsf{m};1}\ (\mathrm{emulate}_{\mathsf{m}}(\mathsf{t})))).\mathbf{1}$$

By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t})\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$. The result follows by Lemma 46.

- rule λ^{u}-Wf-Proj2: We have that

$$\mathrm{emulate}_{\mathsf{m}}(\mathsf{t}.2) = (\mathtt{case}_{\times;\mathtt{m}}\ (\mathrm{upgrade}_{\mathsf{m};1}\ (\mathrm{emulate}_{\mathsf{m}}(\mathsf{t})))).\mathbf{2}$$

By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t})\ \square_{\mathsf{n}}\ \mathsf{t} : \mathtt{EmulDV}_{\mathsf{m;p}}$. The result follows by Lemma 46.

- rule λ^{u}-Wf-Case: We have that

$$\begin{aligned}&\mathrm{emulate}_{\mathsf{m}}(\mathrm{case}\ \mathsf{t}_1\ \mathrm{of}\ \mathrm{inl}\ \mathsf{x} \mapsto \mathsf{t}_2 \mid \mathrm{inr}\ \mathsf{x} \mapsto \mathsf{t}_3) = \\ &\mathrm{case}\ \mathtt{case}_{\uplus;\mathtt{m}}\ (\mathrm{upgrade}_{\mathsf{m};1}\ (\mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_1)))\ \mathrm{of}\ \mathrm{inl}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_2) \mid \mathrm{inr}\ \mathbf{x} \mapsto \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_3)\end{aligned}$$

By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_1)\ \square_{\mathsf{n}}\ \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$, $\mathtt{toEmul}(\Gamma, \mathsf{x}])_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_2)\ \square_{\mathsf{n}}\ \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}$ and $\mathtt{toEmul}(\Gamma, \mathsf{x}])_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_3)\ \square_{\mathsf{n}}\ \mathsf{t}_3 : \mathtt{EmulDV}_{\mathsf{m;p}}$. The result follows by Lemma 43.

- rule λ^{u}-Wf-Wrong: We have that $\mathrm{emulate}_{\mathsf{m}}(\mathsf{wrong}) = \mathrm{omega}_{\mathrm{UVal}_{\mathbf{m}}}$. So, take $\underline{\mathsf{W}}$ with $\mathsf{lev}(\underline{\mathsf{W}}) \leq n$ and $(\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}]\!]_{\square}$. Then we need to show that $(\underline{\mathsf{W}}, \mathrm{omega}_{\mathrm{UVal}_{\mathbf{m}}}, \mathsf{wrong}) \in \mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$. This follows easily by Lemma 6 and the definition of $\mathcal{E}[\![\mathtt{EmulDV}_{\mathsf{m;p}}]\!]_{\square}$.

- rule λ^{u}-Wf-If We have that

$$\begin{aligned}&\mathrm{emulate}_{\mathsf{m}}(\mathrm{if}\ \mathsf{t}_1\ \mathrm{then}\ \mathsf{t}_2\ \mathrm{else}\ \mathsf{t}_3) = \\ &\mathrm{if}\ (\mathtt{case}_{\mathtt{Bool};\mathtt{n}}(\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}\mathsf{t}_1)))\ \mathrm{then}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2)\ \mathrm{else}\ \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_3)\end{aligned}$$

By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_1)\ \square_{\mathsf{n}}\ \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$, $\mathtt{toEmul}(\Gamma, \mathsf{x}])_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_2)\ \square_{\mathsf{n}}\ \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}$ and $\mathtt{toEmul}(\Gamma, \mathsf{x}])_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_3)\ \square_{\mathsf{n}}\ \mathsf{t}_3 : \mathtt{EmulDV}_{\mathsf{m;p}}$. The result follows by Lemma 47.

- rule λ^{u}-Wf-Seq We have that

$$\mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_1; \mathsf{t}_2) = \\ (\mathtt{case}_{\mathtt{Unit;n}}\ (\mathrm{upgrade}_{\mathsf{n};1}(\mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_1)))); \mathrm{emulate}_{\mathsf{n}}(\mathsf{t}_2)$$

  By induction, we have that $\mathtt{toEmul}(\Gamma)_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_1) \,\square_{\mathsf{n}}\, \mathsf{t}_1 : \mathtt{EmulDV}_{\mathsf{m;p}}$, $\mathtt{toEmul}(\Gamma, \mathsf{x}])_{\mathsf{m;p}} \vdash \mathrm{emulate}_{\mathsf{m}}(\mathsf{t}_2) \,\square_{\mathsf{n}}\, \mathsf{t}_2 : \mathtt{EmulDV}_{\mathsf{m;p}}$. The result follows by Lemma 48.

$\square$

Theorem 12 (Emulate is semantics preserving for contexts)**.** *If* $\vdash \mathfrak{C} : \Gamma' \to \Gamma$, *if* $(m > n$ *and* $p = \mathtt{precise})$ *or* $(\square = \lesssim$ *and* $p = \mathtt{imprecise})$, *then* $\vdash \mathrm{emulate}_{\mathsf{m}}(\mathfrak{C}) \,\square_{\mathsf{n}}\, \mathfrak{C} : \mathtt{toEmul}(\Gamma')_{\mathsf{m;p}}, \mathtt{EmulDV}_{\mathsf{m;p}} \to \mathtt{toEmul}(\Gamma)_{\mathsf{m;p}}, \mathtt{EmulDV}_{\mathsf{m;p}}$

Proof. We prove this by induction on the judgement $\vdash \mathfrak{C} : \Gamma' \to \Gamma$.

- rule λ^{u}-Wf-Ctx-Hole Follows trivially.
- rule λ^{u}-Wf-Ctx-Lam Follows by the induction hypothesis and Lemma 41.
- rule λ^{u}-Wf-Ctx-Pair1 Follows by the induction hypothesis and by Theorem 11 and Lemma 44.
- rule λ^{u}-Wf-Ctx-Pair2 Follows by the induction hypothesis and by Theorem 11 and Lemma 44.
- rule λ^{u}-Wf-Ctx-Inl Follows by the induction hypothesis and by Lemma 45.
- rule λ^{u}-Wf-Ctx-Inr Follows by the induction hypothesis and by Lemma 45.
- rule λ^{u}-Wf-Ctx-App1 Follows by the induction hypothesis and by Theorem 11 and Lemma 42.
- rule λ^{u}-Wf-Ctx-App2 Follows by the induction hypothesis and by Theorem 11 and Lemma 42.
- rule λ^{u}-Wf-Ctx-Proj1 Follows by the induction hypothesis and by Lemma 46.
- rule λ^{u}-Wf-Ctx-Proj2 Follows by the induction hypothesis and by Lemma 46.
- rule λ^{u}-Wf-Ctx-Case1 Follows by the induction hypothesis and by Theorem 11 and Lemma 43.
- rule λ^{u}-Wf-Ctx-Case2 Follows by the induction hypothesis and by Theorem 11 and Lemma 43.
- rule λ^{u}-Wf-Ctx-Case3 Follows by the induction hypothesis and by Theorem 11 and Lemma 43.
- rule λ^{u}-Type-Ctx-If1 Follows by the induction hypothesis and by Theorem 11 and Lemma 47.

- rule λ^{u}-Type-Ctx-If2 Follows by the induction hypothesis and by Theorem 11 and Lemma 47.
- rule λ^{u}-Type-Ctx-If3 Follows by the induction hypothesis and by Theorem 11 and Lemma 47.
- rule λ^{u}-Type-Ctx-Seq1 Follows by the induction hypothesis and by Theorem 11 and Lemma 48.
- rule λ^{u}-Type-Ctx-Seq2 Follows by the induction hypothesis and by Theorem 11 and Lemma 48.

$\square$

6.6 Approximate back-translation

The n-approximate back-translation of a context $\mathfrak{C}$ with a hole of type τ is defined as follows.

$$\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}} \stackrel{\mathsf{def}}{=} \mathrm{emulate}_{\mathsf{n}+1}(\mathfrak{C})[\mathbf{inject}_{\tau;\mathsf{n}}\ \cdot]$$

Lemma 49 (Correctness of $\langle\!\langle\cdot\rangle\!\rangle_{\tau;\mathsf{n}}$)**.** *If ($m \geq n$ and $p = \mathtt{precise}$) or ($\square = \lesssim$ and $p = \mathtt{imprecise}$), then* $\vdash \mathfrak{C} : \emptyset \to \emptyset$ *and* $\emptyset \vdash \mathbf{t}\ \square_{\mathsf{n}}\ \mathsf{t} : \tau$ *implies* $\emptyset \vdash \langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{m}}[\mathbf{t}]\ \square_{\mathsf{n}}\ \mathfrak{C}[\mathsf{protect}_\tau\ \mathsf{t}] : \mathtt{EmulDV}_{\mathsf{m};\mathsf{p}}$.

Proof. Follows from Theorems 10 and 12 $\square$

6.7 Contextual equivalence preservation

Theorem 13. *If* $\emptyset \vdash \mathbf{t_1} : \tau$, $\emptyset \vdash \mathbf{t_2} : \tau$ *and* $\emptyset \vdash \mathbf{t_1} \simeq_{ctx} \mathbf{t_2} : \tau$, *then* $\emptyset \vdash \mathsf{protect}_\tau(\mathtt{erase}(\mathbf{t_1})) \simeq_{ctx} \mathsf{protect}_\tau(\mathtt{erase}(\mathbf{t_1}))$.

Proof. Note that $\mathsf{protect}_\tau\ (\mathtt{erase}(\mathbf{t_1})) = [\![\mathbf{t_1}]\!]$ by definition and similarly for $\mathbf{t_2}$.

Take a $\vdash \mathfrak{C} : \emptyset \to \emptyset$ and suppose that $\mathfrak{C}[\mathsf{protect}_\tau(\mathtt{erase}(\mathbf{t_1}))]\Downarrow$, then by symmetry, it suffices to show that $\mathfrak{C}[\mathsf{protect}_\tau(\mathtt{erase}(\mathbf{t_2}))]\Downarrow$.

Take n strictly larger than the number of steps in the termination of $\mathfrak{C}[\mathsf{protect}_\tau(\mathtt{erase}(\mathbf{t_1}))]\Downarrow$.

By Theorem 4, we have that $\emptyset \vdash \mathbf{t_1} \gtrsim_{\mathsf{n}} \mathtt{erase}(\mathbf{t_1}) : \tau$.

By Lemma 49, we then have (taking $m = n \geq n$, $p = \mathtt{precise}$ and $\square = \gtrsim$) that

$$\emptyset \vdash \langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\mathbf{t_1}] \gtrsim_{\mathsf{n}} \mathfrak{C}[\mathsf{protect}_\tau\ (\mathtt{erase}(\mathbf{t_1}))] : \mathtt{EmulDV}_{\mathsf{n};\mathtt{precise}}.$$

Now by Lemma 15, by $\mathfrak{C}[\mathsf{protect}_\tau(\mathtt{erase}(\mathbf{t_1}))]\Downarrow$, and by the choice of n, we have that $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\mathbf{t_1}]\Downarrow$.

It now follows from $\emptyset \vdash \mathbf{t_1} \simeq_{ctx} \mathbf{t_2} : \tau$ and $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\mathbf{t_1}]\Downarrow$ that $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\mathbf{t_2}]\Downarrow$.

Now take n' the number of steps in the termination of $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\mathbf{t_2}]\Downarrow$. We have from Theorem 4 that $\emptyset \vdash \mathbf{t_2} \lesssim_{\mathsf{n'}} \mathtt{erase}(\mathbf{t_2}) : \tau$.

By Lemma 49, we then have (taking $m = n$, $n = n'$, $p = \mathtt{imprecise}$ and $\square = \lesssim$) that

$$\emptyset \vdash \langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\mathbf{t_2}] \lesssim_{\mathsf{n'}} \mathfrak{C}[\mathsf{protect}_\tau\ (\mathtt{erase}(\mathbf{t_2}))] : \mathtt{EmulDV}_{\mathsf{n};\mathtt{imprecise}}$$

Now by Lemma 14, by $\langle\langle \mathfrak{C} \rangle\rangle_{\tau;\mathsf{n}}[\mathbf{t_2}]\Downarrow$, and by the choice of n', we have that $\mathfrak{C}[\mathsf{protect}_{\tau}\ (\texttt{erase}(\mathbf{t_2}))]\Downarrow$ as required. $\square$

7 Compiler full abstraction

Theorem 14 ($[\![\cdot]\!]$ is fully-abstract)**.** *If* $\emptyset \vdash \mathbf{t_1} : \tau$, $\emptyset \vdash \mathbf{t_2} : \tau$ *then* $\emptyset \vdash \mathbf{t_1} \simeq_{ctx} \mathbf{t_2} : \tau$ *iff* $\emptyset \vdash \mathsf{protect}_{\tau}(\mathtt{erase}(\mathbf{t_1})) \simeq_{ctx} \mathsf{protect}_{\tau}(\mathtt{erase}(\mathbf{t_1}))$.

Proof. Combine Theorems 16 and 17. □

8 Modular Full Abstraction

8.1 Linking

If

$$\mathbf{x_2} : \tau_2' \to \tau_2 \vdash \mathbf{t_1} : \tau_1' \to \tau_1$$
$$\mathbf{x_1} : \tau_1' \to \tau_1 \vdash \mathbf{t_2} : \tau_2' \to \tau_2$$

then

$$\mathbf{t_1} + \mathbf{t_2} \stackrel{\mathsf{def}}{=} \left(\begin{array}{l} \mathrm{fix}_{\mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2))} \\ \quad (\lambda p : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)).\, \lambda\_ : \mathtt{Unit}. \\ \qquad\qquad \left\langle \begin{array}{l} \lambda x_1' : \tau_1'.\, ((\lambda x_2 : \tau_2' \to \tau_2.\, t_1)\ ((p\ \mathtt{unit}).2))\ x_1', \\ \quad \lambda x_2' : \tau_2'.\, ((\lambda x_1 : \tau_1' \to \tau_1.\, t_2)\ ((p\ \mathtt{unit}).1))\ x_2' \end{array} \right\rangle) \end{array} \right) \mathtt{unit}$$

We can show that the this produces a well-typed term:

$$(\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}) + (\lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}) : ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2))$$

If

$$\mathsf{x}_2 \vdash \mathsf{t}_1$$
$$\mathsf{x}_1 \vdash \mathsf{t}_2$$

then

$$\mathsf{t}_1 + \mathsf{t}_2 \stackrel{\mathsf{def}}{=} \left(\mathit{fix} \left(\lambda p.\, \lambda\_.\, \left\langle \begin{array}{l} \lambda x_1'.\, ((\lambda x_2.\, t_1)\ (p\ \mathrm{unit}).2)\ x_1', \\ \quad \lambda x_2'.\, ((\lambda x_1.\, t_2)\ (p\ \mathrm{unit}).1)\ x_2' \end{array} \right\rangle \right) \right)\ \mathtt{unit}$$

8.2 Compiler

The compiler changes as follows, provided that $\mathbf{x_2} : \tau_2' \to \tau_2 \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} : \tau_1' \to \tau_1$, then:

$$[\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} = \mathsf{protect}_{\tau_1' \to \tau_1}(\lambda \mathsf{x}_1'.\, ((\lambda \mathsf{x}_2.\, \mathtt{erase}(\mathbf{t_1}))(\mathsf{confine}_{\tau_2' \to \tau_2}\ \mathsf{x}_2)))$$

8.3 Additional Theorems and Proofs

This section presents which additional theorems are needed for modular full abstraction and which theorems replace which old ones.

Lemma 50 (An extra confine is just fine)**.** *If*

- $\mathbf{\Gamma}, \mathbf{x} : \tau' \vdash \mathbf{t} : \tau$ *(Ht),*

then $\mathbf{\Gamma}, \mathbf{x} : \tau' \vdash \mathbf{t} \ \square_n\ (\lambda \mathsf{x}.\, \mathtt{erase}(\mathbf{t_1}))(\mathsf{confine}_{\tau'}\ \mathsf{x}) : \tau$

Proof. By Definition 5 we need to prove forall n:

$$\forall \underline{\mathsf{W}}.\ \mathsf{lev}(\underline{\mathsf{W}}) \leq n \Rightarrow \forall (\underline{\mathsf{W}}, \gamma, \gamma) \in \mathcal{G}[\![\mathbf{\Gamma}, \mathbf{x} : \tau']\!]_{\square}.$$
$$(\underline{\mathsf{W}}, \mathbf{t}\gamma, (\lambda \mathsf{x}.\ \mathtt{erase}(\mathbf{t}))(\mathsf{confine}_{\tau'}\ \mathsf{x}))\gamma) \in \mathcal{E}[\![\tau]\!]_{\square}$$

Take γ and γ to be $[\mathbf{v}/\mathbf{x}]\gamma'$ and $[\mathsf{v}/\mathsf{x}]\gamma'$ respectively.
So $(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}) \in \mathcal{V}[\![\tau']\!]$ (Hv) and $(\underline{\mathsf{W}}, \gamma', \gamma') \in \mathcal{G}[\![\mathbf{\Gamma}]\!]$ (Hg).
The thesis is:

$$\forall \underline{\mathsf{W}}.\ \mathsf{lev}(\underline{\mathsf{W}}) \leq n \Rightarrow$$
$$(\underline{\mathsf{W}}, \mathbf{t}[\mathbf{v}/\mathbf{x}]\gamma', (\lambda \mathsf{x}.\ \mathtt{erase}(\mathbf{t}))(\mathsf{confine}_{\tau'}\ \mathsf{x})[\mathsf{v}/\mathsf{x}]\gamma') \in \mathcal{E}[\![\tau]\!]_{\square}$$

so

$$\forall \underline{\mathsf{W}}.\ \mathsf{lev}(\underline{\mathsf{W}}) \leq n \Rightarrow$$
$$(\underline{\mathsf{W}}, \mathbf{t}[\mathbf{v}/\mathbf{x}]\gamma', (\lambda \mathsf{x}.\ \mathtt{erase}(\mathbf{t}))(\mathsf{confine}_{\tau'}\ \mathsf{v})\gamma') \in \mathcal{E}[\![\tau]\!]_{\square}$$

By Lemma 33 and Hv, we have that

$$\begin{aligned}&(\lambda \mathsf{x}.\ \mathtt{erase}(\mathbf{t}))(\mathsf{confine}_{\tau'}\ \mathsf{v})\gamma' \\ \hookrightarrow &(\lambda \mathsf{x}.\ \mathtt{erase}(\mathbf{t}))(\mathsf{v}')\gamma'\end{aligned}$$

and that (Hvpp)

$$(\underline{\mathsf{W}}, \mathbf{v}, \mathsf{v}') \in \mathcal{V}[\![\tau']\!]$$

So we know that:

$$\begin{aligned}&(\lambda \mathsf{x}.\ \mathtt{erase}(\mathbf{t}))\ (\mathsf{confine}_{\tau'}\ \mathsf{v})\gamma' \\ \hookrightarrow &(\lambda \mathsf{x}.\ \mathtt{erase}(\mathbf{t}))\ \mathsf{v}'\gamma' \\ \hookrightarrow &\mathtt{erase}(\mathbf{t})[\mathsf{v}'/\mathsf{x}]\gamma'\end{aligned}$$

By Lemma 8, it suffices to prove that

$$(\underline{\mathsf{W}}, \mathbf{t}[\mathbf{v}/\mathbf{x}]\gamma', \mathtt{erase}(\mathbf{t})[\mathsf{v}'/\mathsf{x}]\gamma') \in \mathcal{E}[\![\tau]\!]_{\square}$$

By Theorem 5 with Ht we know that (Htr)

$$\mathbf{\Gamma}, \mathbf{x} : \tau' \vdash \mathbf{t} \sqsupset_n \mathtt{erase}(\mathbf{t}) : \tau$$

By Definition 5 we get

$$\forall \underline{\mathsf{W}'}.\ \mathsf{lev}(\underline{\mathsf{W}'}) \leq n \Rightarrow \forall (\underline{\mathsf{W}'}, \gamma'', \gamma'') \in \mathcal{G}[\![\mathbf{\Gamma}, \mathbf{x} : \tau']\!]_{\square}.$$
$$(\underline{\mathsf{W}'}, \mathbf{t}\gamma'', \mathtt{erase}(\mathbf{t})\gamma'') \in \mathcal{E}[\![\tau]\!]_{\square}$$

We instantiate $\underline{\mathsf{W}'}$ with $\underline{\mathsf{W}}$, γ'' with $[\mathbf{v}/\mathbf{x}]\gamma'$ and γ'' with $[\mathsf{v}'/\mathsf{x}]\gamma'$
By Hvpp and Hg we have that $(\underline{\mathsf{W}'}, [\mathbf{v}/\mathbf{x}]\gamma', [\mathsf{v}'/\mathsf{x}]\gamma') \in \mathcal{G}[\![\mathbf{\Gamma}, \mathbf{x} : \tau']\!]_{\square}$.
So the thesis holds. $\square$

Theorem 15 (Confining free variables is correct (aka, $[\![\cdot]\!]^{\lambda^\tau}_{\lambda^u}$ is correct))**.** *If*

- $\mathbf{x_2} : \tau_2' \to \tau_2 \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} : \tau_1' \to \tau_1$ *(Ht),*

then $\mathbf{x_2} : \tau_2' \to \tau_2 \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} \;\square_n\; \mathsf{protect}_{\tau_1' \to \tau_1}(\lambda \mathsf{x}_1'.\, ((\lambda \mathsf{x}_2.\, \mathtt{erase}(\mathbf{t_1}))(\mathsf{confine}_{\tau_2' \to \tau_2}\ \mathsf{x}_2))) : \tau_1' \to \tau_1$

Proof. By Theorem 6 it is sufficient to prove that
$\mathbf{x_2} : \tau_2' \to \tau_2 \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} \;\square_n\; (\lambda \mathsf{x}_1'.\, ((\lambda \mathsf{x}_2.\, \mathtt{erase}(\mathbf{t_1}))(\mathsf{confine}_{\tau_2' \to \tau_2}\ \mathsf{x}_2))) : \tau_1' \to \tau_1$
By Lemma 21 it suffices to prove that:
$\mathbf{x_2} : \tau_2' \to \tau_2 ; \mathbf{x_1'} : \tau_1' \vdash \mathbf{t_1} \;\square_n\; ((\lambda \mathsf{x}_2.\, \mathtt{erase}(\mathbf{t_1}))(\mathsf{confine}_{\tau_2' \to \tau_2}\ \mathsf{x}_2)) : \tau_1$
This holds by Lemma 50.

□

Theorem 16 ($[\![\cdot]\!]^{\lambda^\tau}_{\lambda^u}$ reflects equivalence)**.** *If*

- $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} : \tau_1' \to \tau_1$ *(Ht1),*
- $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x_2'} : \tau_1'.\, \mathbf{t_2} : \tau_1' \to \tau_1$ *(Ht2),*
- $\mathsf{x} \vdash [\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} \simeq_{ctx} [\![\lambda \mathbf{x_2'} : \tau_1'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}$ *(Htc),*

then $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} \simeq_{ctx} \lambda \mathbf{x_2'} : \tau_1'.\, \mathbf{t_2} : \tau_1' \to \tau_1$.

Proof. In the following we shorten $\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}$ to $\mathbf{t_1}$ and $\lambda \mathbf{x_2'} : \tau_1'.\, \mathbf{t_2}$ to $\mathbf{t_2}$.
Take $\mathfrak{C}$ so that $\vdash \mathfrak{C} : \mathbf{x} : \tau' \to \tau, \tau_1' \to \tau_1 \to \emptyset, \tau''$ (Hk).
We need to prove that $\mathfrak{C}[\mathbf{t_1}]\Downarrow$ iff $\mathfrak{C}[\mathbf{t_2}]\Downarrow$.
By symmetry, it suffices to prove the $\Rightarrow$ direction.
So assume that $\mathfrak{C}[\mathbf{t_1}]\Downarrow$ (Ht1d).
Then we need to prove that $\mathfrak{C}[\mathbf{t_2}]\Downarrow$.
Define $\mathfrak{C} \overset{\mathsf{def}}{=} \mathtt{erase}(\mathfrak{C})$.
Theorem 5 tells us that $\vdash \mathfrak{C} \;\square_n\; \mathfrak{C} : \mathbf{x} : \tau' \to \tau, \tau_1' \to \tau_1 \to \emptyset, \tau''$.
Theorem 15 with Ht1 yields $\mathbf{x} : \tau' \to \tau \vdash \mathbf{t_1} \;\square_n\; [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} : \tau$ (Ht1c).
Theorem 15 with Ht2 yields $\mathbf{x} : \tau' \to \tau \vdash \mathbf{t_2} \;\square_n\; [\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u} : \tau$ (Ht2c).
By definition of $\vdash \mathfrak{C} \;\square_n\; \mathfrak{C} : \mathbf{x} : \tau' \to \tau, \tau_1' \to \tau_1 \to \emptyset, \tau''$ with Ht1c and Ht2c, we get that

- $\emptyset \vdash \mathfrak{C}[\mathbf{t_1}] \;\square_n\; \mathfrak{C}[[\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u}] : \tau''$ (Ht1r) and
- $\emptyset \vdash \mathfrak{C}[\mathbf{t_2}] \;\square_n\; \mathfrak{C}[[\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}] : \tau''$ (Ht2r).

By Lemma 16 with Ht1d and Ht1r imply that $\mathfrak{C}[[\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$ (Hk1).
By Lemma 18 with Hk we get $\vdash \mathfrak{C} : \mathsf{x} \to \emptyset$.
So, from Htc and Hk1, we get that $\mathfrak{C}[[\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$ (Ht2t).
By Lemma 16 with Ht2r and Ht2t we now get that $\mathfrak{C}[\mathbf{t_2}]\Downarrow$ □

Theorem 17 ($[\![\cdot]\!]^{\lambda^\tau}_{\lambda^u}$ preserves equivalence)**.** *If*

- $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1} : \tau'_1 \to \tau_1$ *(Ht1)*,
- $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x}'_2 : \tau'_1 . \mathbf{t_2} : \tau'_1 \to \tau_1$ *(Ht2)*,
- $\mathbf{x} : \tau' \vdash \lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1} \simeq_{ctx} \lambda \mathbf{x}'_1 : \tau'_2 . \mathbf{t_2} : \tau$ *(Htc)*,

then $\mathsf{x} \vdash [\![\lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}} \simeq_{ctx} [\![\lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}}$.

Proof. Take $\mathsf{a} \vdash \mathfrak{C} : \mathsf{x} \to \emptyset$.

Assume that $\mathfrak{C}[[\![\lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}}]\Downarrow$ (Ht1d).

By symmetry, it suffices to show that $\mathfrak{C}[[\![\lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}}]\Downarrow$.

Take n strictly larger than the number of steps in the termination of $\mathfrak{C}[[\![\lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}}]\Downarrow$.

By Theorem 15 with Ht1 we have that $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1} \gtrsim_{\mathsf{n}} [\![\lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}} : \tau'_1 \to \tau_1$.

By Lemma 49 , taking $m = n$, so $m \geq n$ and $p = \texttt{precise}$ and $\square = \gtrsim$, we then have that $\emptyset \vdash \langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1}] \gtrsim_{\mathsf{n}} \mathfrak{C}[[\![\lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}}] : \mathtt{EmulDV}_{\mathsf{n};\mathtt{precise}}$.

By Lemma 15 with Ht1d, and by the choice of n, we have that $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1}]\Downarrow$ (Ht1t).

From Htc and Ht1t we have that $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2}]\Downarrow$.

Take n' the number of steps in the termination of $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2}]\Downarrow$ (Ht2t).

From Theorem 15 with Ht2 we have that $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2} \lesssim_{\mathsf{n}'} [\![\lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}} : \tau$.

By Lemma 49 , taking $m = n$, $n = n'$, $p = \texttt{imprecise}$ and $\square = \lesssim$ we then have that $\emptyset \vdash \langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2}] \lesssim_{\mathsf{n}'} \mathfrak{C}[[\![\lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}}] : \mathtt{EmulDV}_{\mathsf{n};\mathtt{imprecise}}$

By Lemma 14 with Ht2t, and by the choice of n', we have that $\mathfrak{C}[[\![\lambda \mathbf{x}'_2 : \tau'_2 . \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}}]\Downarrow$. $\square$

Theorem 18 (Compiler Full Abstraction)**.**

- $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x}'_1 : \tau'_1 . \mathbf{t_1} : \tau'_1 \to \tau_1$ *(Ht1)*,
- $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x}'_2 : \tau'_1 . \mathbf{t_2} : \tau'_1 \to \tau_1$ *(Ht2)*,

then $\mathbf{x} : \tau' \to \tau \vdash \lambda \mathbf{x}'_1 . \mathbf{t_1} \simeq_{ctx} \lambda \mathbf{x}'_2 . \mathbf{t_2} : \tau'_1 \to \tau_1 \iff \mathsf{x} \vdash [\![\lambda \mathbf{x}'_1 . \mathbf{t_1}]\!]^{\mathcal{S}}_{\mathcal{T}} \simeq_{ctx} [\![\lambda \mathbf{x}'_2 . \mathbf{t_2}]\!]^{\mathcal{S}}_{\mathcal{T}}$.

Proof. By Theorem 17 and Theorem 16. $\square$

8.3.1 Proofs about Modularity

Lemma 51 (Source linking is related to target liking)**.** *If*

- $\mathbf{x_2} : \tau'_2 \to \tau_2 \vdash \mathbf{t_1} : \tau'_1 \to \tau_1$ *(Ht1)*
- $\mathbf{x_1} : \tau'_1 \to \tau_1 \vdash \mathbf{t_2} : \tau'_2 \to \tau_2$ *(Ht2)*

then $\emptyset \vdash \mathbf{t_1} + \mathbf{t_2} \;\square_n\; [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}} + [\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^\mathsf{u}} : (\tau'_1 \to \tau_1) \times (\tau'_2 \to \tau_2)$.

Proof. Unfold the definitions of linking. We need to prove that:

$$\emptyset \vdash \left(\begin{pmatrix} \text{fix}_{\texttt{Unit}\to((\tau_1'\to\tau_1)\times(\tau_2'\to\tau_2))} \\ (\lambda p : \texttt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)).\, \lambda\_ : \texttt{Unit}. \\ \langle \lambda x_1' : \tau_1'.\, ((\lambda x_2 : \tau_2' \to \tau_2.\, t_1)\ ((p\ \texttt{unit}).2))\ x_1', \\ \lambda x_2' : \tau_2'.\, ((\lambda x_1 : \tau_1' \to \tau_1.\, t_2)\ ((p\ \texttt{unit}).1))\ x_2' \rangle) \end{pmatrix}\ \texttt{unit}\right)$$

$$\square_n$$

$$\left(\left(\mathit{fix}\left(\lambda p.\, \lambda\_.\, \left\langle \begin{matrix} \lambda x_1'.\, ((\lambda x_2.\, [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u})\ (p\ \text{unit}).2)\ x_1', \\ \lambda x_2'.\, ((\lambda x_1.\, [\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u})\ (p\ \text{unit}).1)\ x_2' \end{matrix} \right\rangle\right)\right)\ \texttt{unit}\right)$$

$$: (\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})$$

By Lemma 23 it suffices to show the following:

- $$\emptyset \vdash \begin{pmatrix} \text{fix}_{\texttt{Unit}\to((\tau_1'\to\tau_1)\times(\tau_2'\to\tau_2))} \\ (\lambda p : \texttt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)).\, \lambda\_ : \texttt{Unit}. \\ \langle \lambda x_1' : \tau_1'.\, ((\lambda x_2 : \tau_2' \to \tau_2.\, t_1)\ ((p\ \texttt{unit}).2))\ x_1', \\ \lambda x_2' : \tau_2'.\, ((\lambda x_1 : \tau_1' \to \tau_1.\, t_2)\ ((p\ \texttt{unit}).1))\ x_2' \rangle) \end{pmatrix}$$

  $$\square_n$$

  $$\left(\mathit{fix}\left(\lambda \mathsf{p}.\, \lambda\_.\, \left\langle \begin{matrix} \lambda x_1'.\, ((\lambda x_2.\, [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u})\ (p\ \text{unit}).2)\ x_1', \\ \lambda x_2'.\, ((\lambda x_1.\, [\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u})\ (p\ \text{unit}).1)\ x_2' \end{matrix} \right\rangle\right)\right)$$

  $$: \texttt{Unit} \to (\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})$$

  By Lemma 31 it suffices to show that:

  $$\emptyset \vdash \begin{pmatrix} (\lambda p : \texttt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)).\, \lambda\_ : \texttt{Unit}. \\ \langle \lambda x_1' : \tau_1'.\, ((\lambda x_2 : \tau_2' \to \tau_2.\, t_1)\ ((p\ \texttt{unit}).2))\ x_1', \\ \lambda x_2' : \tau_2'.\, ((\lambda x_1 : \tau_1' \to \tau_1.\, t_2)\ ((p\ \texttt{unit}).1))\ x_2' \rangle) \end{pmatrix}$$

  $$\square_n$$

  $$\left(\lambda \mathsf{p}.\, \lambda\_.\, \left\langle \begin{matrix} \lambda x_1'.\, ((\lambda x_2.\, [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u})\ (p\ \text{unit}).2)\ x_1', \\ \lambda x_2'.\, ((\lambda x_1.\, [\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u})\ (p\ \text{unit}).1)\ x_2' \end{matrix} \right\rangle\right)$$

  $$: (\texttt{Unit} \to (\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})) \to (\texttt{Unit} \to (\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}}))$$

By Lemma 21 it suffices to show that:

$$\begin{array}{l}
\mathbf{p} : \mathtt{Unit} \to ((\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})) \vdash \\
\left(\begin{array}{l} (\lambda_ : \mathtt{Unit}. \\ \ \langle \lambda x_1' : \tau_1'. ((\lambda x_2 : \tau_2' \to \tau_2. t_1)\ ((p\ \mathtt{unit}).2))\ x_1', \\ \ \lambda x_2' : \tau_2'. ((\lambda x_1 : \tau_1' \to \tau_1. t_2)\ ((p\ \mathtt{unit}).1))\ x_2' \rangle) \end{array}\right) \\
\square_n \\
\left(\lambda_. \left\langle \begin{array}{l} \lambda x_1'. ((\lambda x_2. [\![\mathbf{t_1}]\!]_{\lambda^{u}}^{\lambda^{\tau}})\ (p\ \text{unit}).2)\ x_1', \\ \lambda x_2'. ((\lambda x_1. [\![\mathbf{t_2}]\!]_{\lambda^{u}}^{\lambda^{\tau}})\ (p\ \text{unit}).1)\ x_2' \end{array} \right\rangle\right) \\
: \mathtt{Unit} \to (\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})
\end{array}$$

By Lemma 21 it suffices to show that:

$$\begin{array}{l}
\mathbf{p} : \mathtt{Unit} \to ((\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})) \vdash \\
\left(\begin{array}{l} \langle \lambda x_1' : \tau_1'. ((\lambda x_2 : \tau_2' \to \tau_2. t_1)\ ((p\ \mathtt{unit}).2))\ x_1', \\ \lambda x_2' : \tau_2'. ((\lambda x_1 : \tau_1' \to \tau_1. t_2)\ ((p\ \mathtt{unit}).1))\ x_2' \rangle) \end{array}\right) \\
\square_n \\
\left\langle \begin{array}{l} \lambda x_1'. ((\lambda x_2. [\![\mathbf{t_1}]\!]_{\lambda^{u}}^{\lambda^{\tau}})\ (p\ \text{unit}).2)\ x_1', \\ \lambda x_2'. ((\lambda x_1. [\![\mathbf{t_2}]\!]_{\lambda^{u}}^{\lambda^{\tau}})\ (p\ \text{unit}).1)\ x_2' \end{array} \right\rangle \\
: (\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})
\end{array}$$

By Lemma 22 it suffices to show that:

- $$\begin{array}{l}
\mathbf{p} : \mathtt{Unit} \to ((\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})) \vdash \\
\lambda \mathbf{x}_{\mathbf{1}}' : \tau_{\mathbf{1}}'. ((\lambda \mathbf{x_2} : \tau_{\mathbf{2}}' \to \tau_{\mathbf{2}}. \mathbf{t_1})\ ((\mathbf{p}\ \mathtt{unit}).\mathbf{2}))\ \mathbf{x}_{\mathbf{1}}' \\
\square_n \\
\lambda \mathsf{x}_1'. ((\lambda \mathsf{x}_2. [\![\mathbf{t_1}]\!]_{\lambda^{u}}^{\lambda^{\tau}})\ (\mathsf{p}\ \text{unit}).2)\ \mathsf{x}_1' \\
: (\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}})
\end{array}$$

  By Lemma 21 it suffices to show that:

  $$\begin{array}{l}
\mathbf{p} : \mathtt{Unit} \to ((\tau_{\mathbf{1}}' \to \tau_{\mathbf{1}}) \times (\tau_{\mathbf{2}}' \to \tau_{\mathbf{2}})); \mathbf{x}_{\mathbf{1}}' : \tau_{\mathbf{1}}' \vdash \\
((\lambda \mathbf{x_2} : \tau_{\mathbf{2}}' \to \tau_{\mathbf{2}}. \mathbf{t_1})\ ((\mathbf{p}\ \mathtt{unit}).\mathbf{2}))\ \mathbf{x}_{\mathbf{1}}' \\
\square_n \\
((\lambda \mathsf{x}_2. [\![\mathbf{t_1}]\!]_{\lambda^{u}}^{\lambda^{\tau}})\ (\mathsf{p}\ \text{unit}).2)\ \mathsf{x}_1' \\
: \tau_{\mathbf{1}}
\end{array}$$

  By Lemma 23 it suffices to show that:

- $$\begin{array}{l}\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)); \mathbf{x_1'} : \tau_1' \vdash \\ ((\lambda \mathbf{x_2} : \tau_2' \to \tau_2 .\, \mathbf{t_1})\ ((\mathbf{p}\ \mathtt{unit}).\mathbf{2})) \\ \square_n \\ ((\lambda \mathsf{x}_2 .\, [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u})\ (\mathsf{p}\ \mathsf{unit}).2) \\ : \tau_1' \to \tau_1\end{array}$$

  By Lemma 23 it suffices to show that:

  - $$\begin{array}{l}\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)); \mathbf{x_1'} : \tau_1' \vdash \\ (\lambda \mathbf{x_2} : \tau_2' \to \tau_2 .\, \mathbf{t_1}) \\ \square_n \\ (\lambda \mathsf{x}_2 .\, [\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u}) \\ : (\tau_2' \to \tau_2) \to (\tau_1' \to \tau_1)\end{array}$$

    By Lemma 21 it suffices to show:

    $$\begin{array}{l}\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)); \mathbf{x_1'} : \tau_1'; \mathbf{x_2} : \tau_2' \to \tau_2 \vdash \\ (\mathbf{t_1}) \\ \square_n \\ ([\![\mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u}) \\ : (\tau_1' \to \tau_1)\end{array}$$

    This holds by Theorem 15, and weakening, since $\mathbf{p}$ and $\mathbf{x_1'}$ are not in $\mathbf{t_1}$.

  - $$\begin{array}{l}\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)); \mathbf{x_1'} : \tau_1' \vdash \\ (\mathbf{p}\ \mathtt{unit}).\mathbf{2} \\ \square_n \\ (\mathsf{p}\ \mathsf{unit}).2 \\ : \tau_2' \to \tau_2\end{array}$$

    By Lemma 25 it suffices to show:

    $$\begin{array}{l}\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)); \mathbf{x_1'} : \tau_1' \vdash \\ (\mathbf{p}\ \mathtt{unit}) \\ \square_n \\ (\mathsf{p}\ \mathsf{unit}) \\ : (\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)\end{array}$$

By Lemma 23 it suffices to show:

- $$\begin{aligned}&\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)); \mathbf{x_1'} : \tau_1' \vdash \\ &(\mathbf{p}) \\ &\square_n \\ &(\mathsf{p}) \\ &: \mathtt{Unit} \to (\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)\end{aligned}$$

  This holds by definition of the logical relation and by Lemma 10 after the substitutions.

- $$\begin{aligned}&\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)); \mathbf{x_1'} : \tau_1' \vdash \\ &(\mathtt{unit}) \\ &\square_n \\ &(\mathrm{unit}) \\ &: \mathtt{Unit}\end{aligned}$$

  This holds by definition of the value relation for unit.

- $$\begin{aligned}&\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)); \mathbf{x_1'} : \tau_1' \vdash \\ &\mathbf{x_1'} \\ &\square_n \\ &\mathsf{x}_1' \\ &: \tau_1'\end{aligned}$$

  This holds by definition of the logical relation and by Lemma 10 after the substitutions.

- $$\begin{aligned}&\mathbf{p} : \mathtt{Unit} \to ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2)) \vdash \\ &(\quad \lambda x_2' : \tau_2'. ((\lambda x_1 : \tau_1' \to \tau_1. t_2)\ ((p\ \mathtt{unit}).1))\ x_2' \quad) \\ &\square_n \\ &\lambda x_2'. ((\lambda x_1. [\![\mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u})\ (p\ \mathrm{unit}).1)\ x_2' \\ &: (\tau_2' \to \tau_2)\end{aligned}$$

  Analogous to the previous point.

- $$\emptyset \vdash \mathtt{unit} \ \square_n \ \mathtt{unit} : \mathtt{Unit}$$

This holds by definition of the logical relation, and the definition of the value relation for $\mathtt{Unit}$.

□

Theorem 19 (Compiler Modularity)**.** *If*

- $\mathbf{x_2} : \tau_2' \to \tau_2 \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} : \tau_1' \to \tau_1$ *(Ht1)*
- $\mathbf{x_1} : \tau_1' \to \tau_1 \vdash \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2} : \tau_2' \to \tau_2$ *(Ht2)*

then $\emptyset \vdash [\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u} \simeq_{ctx} [\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} + [\![\lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}$.

Proof. $\Rightarrow$ **direction:** Take a $\vdash \mathfrak{C} : \emptyset \to \emptyset$.

Assume that $\mathfrak{C}[[\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$ (Ht1d).

We need to prove that $\mathfrak{C}[[\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} + [\![\lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$.

Take n strictly larger than the number of steps in the termination of $\mathfrak{C}[[\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$.

By Theorem 7 with Ht1 and Ht2 we have that $\emptyset \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2} \gtrsim_{\mathsf{n}} [\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u} : ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2))$.

By Lemma 49 , taking $m = n$, so $m \geq n$ and $p = \mathtt{precise}$ and $\square = \gtrsim$, we then have that $\emptyset \vdash \langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}] \gtrsim_{\mathsf{n}} \mathfrak{C}[[\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}] : \mathtt{EmulDV}_{\mathsf{n};\mathtt{precise}}$.

By Lemma 15 with Ht1d, and by the choice of n, we have that $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\Downarrow$ (Ht1t).

Take n the number of steps in the termination of $\langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\Downarrow$ (Ht2t).

From Lemma 51 we have that $\emptyset \vdash \lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2} \lesssim_{\mathsf{n}} [\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} + [\![\lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u} : ((\tau_1' \to \tau_1) \times (\tau_2' \to \tau_2))$.

By Lemma 49 , taking $m = n$, $p = \mathtt{imprecise}$ and $\square = \lesssim$ we then have that $\emptyset \vdash \langle\!\langle \mathfrak{C} \rangle\!\rangle_{\tau;\mathsf{n}}[\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1} + \lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}] \lesssim_{\mathsf{n'}} \mathfrak{C}[[\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} + [\![\lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}] : \mathtt{EmulDV}_{\mathsf{n};\mathtt{imprecise}}$

By Lemma 14 with Ht2t, and by the choice of n, we have that $\mathfrak{C}[[\![\lambda \mathbf{x_1'} : \tau_1'.\, \mathbf{t_1}]\!]^{\lambda^\tau}_{\lambda^u} + [\![\lambda \mathbf{x_2'} : \tau_2'.\, \mathbf{t_2}]\!]^{\lambda^\tau}_{\lambda^u}]\Downarrow$.

$\Leftarrow$ **direction:** Dual to the previous one.

□

Acknowledgements

Dominique Devriese holds a Postdoctoral mandate from the Research Foundation Flanders (FWO). Marco Patrignani held a Ph.D. fellowship from the Research Foundation Flanders (FWO) during the development of this work. This research is partially funded by project grants from the Research Fund KU Leuven, and from the Research Foundation Flanders (FWO).

References

C.-K. Hur and D. Dreyer. A Kripke logical relation between ML and assembly. In *Principles of Programming Languages*, pages 133–146. ACM, 2011. doi: 10.1145/1926385.1926402.