Compiler Correctness via Contextual Equivalence

Matthew McKay advised by Karl Crary

May 2, 2014

Abstract

We have developed a methodology for verifying the correctness of the closure conversion phase of a compiler, adapted from the work by Perconti and Ahmed. This lets us verify that individual components of programs are compiled correctly, so they can be linked with any other code and still behave as desired. We do this by using a shared language that encompasses both the source and target languages in which the compiled code can be reasoned about alongside its source, which we do using contextual equivalence. Our main improvement over previous methods is that we dont need boundaries that separate the source and target language while inside the shared language.

1 Introduction

In recent years the area of compiler verification has seen significant research attention. Being able to formally verify the correctness of a compiler is a worthwhile goal, as it provides confidence in compilers used for critical systems, which absolutely need their code to be correct. A recent work on verification on which we will focus most of our attention is *Verifying an Open Compiler Using Multi-Language Semantics* [3], by James T. Perconti and Amal Ahmed. Our approach in the following paper is very similar to theirs, however with some improvements, which make up the substantial contributions of our research.

The goal with our research is to verify compilation of program components as opposed to compilation of whole programs. The advantage of this is that realistically, compiled code often gets linked with other code, be it compiled from the same language, a different language, or even just written in the language being compiled to. Thus only being able to verify whole programs severely limits the strength and usefulness of the verification, as we couldn't verify code that gets linked with other code later on (or even just libraries). The problem with checking correctness of components is that they can't be simply run, as they are not complete programs. To do this we make use of contextual equivalence. Two program components are said to be contextually equivalent if, for any potential program context (that is, a program with a hole that the components fit into) that the components could be put in, the two whole programs created by putting those components into the same program context will have the same behavior. To put it simply, the contextual equivalence of two components essentially says that no matter how we use the two components, we have no way to tell them apart. This is desirable, as it fulfills our goal to verify that compiled components can be reliably linked with other code and still behave as intended.

While this sounds great, there is a slight problem. We're compiling code from some source language S to a target language T, which are two different languages. So if we have some term e in the source language that compiles to $\bar{e}$ in the target, these two terms are in different languages, so how can we compare them with contextual equivalence when they can't be put into the same program context?

For our purposes we focused on the closure conversion phase of a compiler, compiling a System F like language, but with the addition of existentials and recursive functions. This is the source language S, and the target language T is the same except instead of recursive functions there are closed recursive functions resulting from closure conversion. Closed recursive functions are simply the same thing as recursive functions, except that they cannot use any variables bound outside of it and is only allowed to use its argument and its own function variable. Given a term

$e : \tau$ in the source language, we compile it to a term in the target language $\bar{e} : |\tau|$, where $|\tau|$ is the type translation of τ.

To solve the previously mentioned problem, we created a new language C that is the combined language of S and T, that is it includes all types and terms of both languages. Thus it is simply the language S but with closed recursive functions as well. This allows us to have program contexts in the combined language, and compare source and target code in the combined language. However a source component may compile to a target component of a different type (they are different languages, after all). To account for this we define two functions in the combined language, $\texttt{over}_\tau$ and $\texttt{back}_\tau$. The function $\texttt{over}_\tau$ takes something in the source language and changes it at the top level to have the appropriate type $|\tau|$ in the target language. Similarly, $\texttt{back}_\tau$ does the opposite, taking something in the target language and changing it so that it has the appropriate type τ in the source language. Of course, these will all be terms in the combined language, since they include structures from both languages. These functions don't actually affect the behavior of the code, as the basically make normal recursive functions look like closed recursive functions and vice versa (and the two types of functions behave in much the same way).

This leads to the statement of our compiler correctness theorem. We prove that, if $\Delta; \Gamma \vdash_S e : \tau$ is in the source language compiles to $\Delta; \Gamma \vdash_T \bar{e'} : |\tau|$ in the target language, then e is contextually equivalent to $\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e'})$ in the combined language, which we write $\Delta; \Gamma \vdash_{\mathcal{C}} e \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e'}) : \tau$. The reason for the substitution (which simply states $\texttt{over}_\tau(x)$ for x in $\bar{e'}$ for every $x : \tau$ in Γ) is that it is necessary to prove the variable case of the theorem. However, we also prove something even stronger. If we know that $\Gamma \vdash_S e_1 \cong e_2 : \tau$ in the source language, then it must be that $\Delta; |\Gamma| \vdash_T \bar{e_1} \cong \bar{e_2} : |\tau|$ in the target language (where $|\Gamma|$ simply means Γ but with the types inside translated). This theorem effectively means that terms that are equivalent in the source language will be equivalent after being compiled, so therefore compiler preserves equivalence. With a little extra work, this theorem will follow from our compiler correctness theorem. The full proofs for these can be found in appendix E.

2 Prior Work

The primary previous work that we are concerned with is that of Perconti and Ahmed [3], which we focused on improving. Their work actually involved three languages, a similar source language, the closure converted language, and an allocation lanugage. Thus their compilation involved two stages, closure conversion and allocation.

In their work they had three languages, F (System F with existentials and recursive types),

C (the closure converted language), and A (the allocation language). They then merge these languages together into a multi-language system that, while it includes everything in each language, still keeps them separate through "boundaries," as they refer to them. These are essentially metafunctions added to the languages that convert from one language to another and allow terms in one language to be placed within a term in another language. Also, since the languages were still distinguished, they didn't share type variables, so special conversions were necessary for polymorphic types when going through boundaries. Their theorem was similar in concept to ours, as it stated that a term in the source was contextually equivalent to the the compiled code, just with the proper boundary function applied to it.

3 The Combined Language

Our language (which well refer to as the "combined language"), which is outlined in Figure 1, is essentially an amplified System F with the addition of existentials and recursive functions (the reason for recursive functions instead of fix will be explained later). The only other thing to note is that along with normal recursive functions, there are what we will refer to as "closed" recursive functions (and correspondingly there is "closed" application to coincide with normal application), which have their own type and are denoted by the hat over them (and similarly, their application uses the same symbol).

$$\tau ::= \alpha \mid \texttt{unit} \mid \texttt{int} \mid \tau \times \tau \mid \tau \to \tau \mid \tau \Rightarrow \tau \mid \forall \alpha.\tau \mid \exists \alpha.\tau$$

$$e ::= \texttt{()} \mid n \mid e\,\texttt{p}\,e \mid \texttt{ifz}(e, e, e) \mid x \mid \langle e, e \rangle \mid \pi_i e$$

$$\mid \texttt{fun}\ f(x : \tau).e \mid e\ e \mid \widehat{\texttt{fun}}\ f(x : \tau).e \mid e\,\widehat{}\,e$$

$$\mid \Lambda \alpha.e \mid e[\tau] \mid \texttt{pack}[\tau', e]\,\texttt{as}\,\exists \alpha.\tau \mid \texttt{unpack}[\alpha, x] = e\,\texttt{in}\,e$$

$$\texttt{p} ::= + \mid - \mid *$$

$$\Gamma ::= \cdot \mid \Gamma, x : \tau \qquad \Delta ::= \cdot \mid \Delta, \alpha$$

Figure 1: The combined language

The reason for having two recursive functions is because this combined language is just that, the combination of two languages, which we will refer to as the "source" language and the "target" language. The source language is System F with existentials and recursive functions and the target language is closure converted System F with existentials and recursive functions, which is simply System F with existentials and closed recursive functions, since the only part of the language that actually changes through closure conversion are the functions and applications. Thus the source and target languages are identical to the combined language, except without the rules relating to the type of function that they do not have. We will denote that a term is typed in the source language by writing $\Delta; \Gamma \vdash_S e : \tau$, and similarly we write $\Delta; \Gamma \vdash_T e : \tau$ for terms typed in the target language. We also write $\Delta; \Gamma \vdash_C e : \tau$ when in the combined language, however since we are primarily discussing the combined language we will forgo its use and just write $\Delta; \Gamma \vdash e : \tau$, unless it is necessary for clarity.

In Figure 2 we have the relevant typing rules for functions and closed functions. The full static and dynamic rules for the combined language can be found in appendices A and B, respectively. It is worth pointing out that the inside of a closed function can only be typed under a context of just the function argument and the bound variable for the function, as they represent functions resulting from closure conversion (hence the name "closed").

$$\frac{\Delta \vdash \tau\, \texttt{type} \quad \Delta; \Gamma, f : \tau \to \tau', x : \tau \vdash e : \tau'}{\Delta; \Gamma \vdash \texttt{fun}\ f(x : \tau).e : \tau \to \tau'}\ Tfun$$

$$\frac{\Delta \vdash \tau\, \texttt{type} \quad \Delta; f : \tau \to \tau', x : \tau \vdash e : \tau'}{\Delta; \Gamma \vdash \widehat{\texttt{fun}}\ f(x : \tau).e : \tau \Rightarrow \tau'}\ Tccfun$$

Figure 2: The static typing rules for functions and closed functions

The purpose for having this combined language is that we can reason about both source terms and target terms together, which is what we want to be able to use contextual equivalence. In the following sections, we will first discuss closure conversion, and then outline contextual equivalence over terms in the combined language and how we actually accomplish it.

4 Closure Conversion

Closure conversion, for our purposes, is very much the standard conversion process. The corresponding type translation $|\tau|$, which is mostly trivial, is in Figure 3. We also define the type translation of a context Γ to be $|\Gamma|$, which is defined by $|\cdot| = \cdot$ and $|\Gamma, x : \tau| = |\Gamma|, x : |\tau|$. This is necessary for stating things about the compiled code, as it can only use variables of the translated types, now that it is in the target language.

$$|\alpha| = \alpha$$

$$|\texttt{unit}| = \texttt{unit}$$

$$|\texttt{int}| = \texttt{int}$$

$$|\tau_1 \times \tau_2| = |\tau_1| \times |\tau_2|$$

$$|\tau_1 \rightarrow \tau_2| = \exists\alpha.((|\tau_1| \times \alpha) \Rightarrow |\tau_2|) \times \alpha$$

$$|\forall\alpha.\tau| = \forall\alpha.|\tau|$$

$$|\exists\alpha.\tau| = \exists\alpha.|\tau|$$

Figure 3: The closure conversion type translation

The translation is of the form $\Delta; \Gamma \vdash_S e \rightsquigarrow \bar{e} : \tau$. The main convenience in analyzing closure conversion is that the source and target language don't differ very much, as most of the conversion doesn't do anything. The only interesting cases involve functions, namely the rules for functions and closure converted functions, which are below in Figure 4. The remaining rules can be found in appendix C.

$$\frac{\begin{array}{cc}\Gamma = x_1 : \tau_1, ..., x_n : \tau_n & \Delta \vdash_S \tau\,\texttt{type} \\ \Delta; \Gamma, x : \tau, f : \tau \rightarrow \tau' \vdash_S e : \tau' \rightsquigarrow \bar{e} & \tau_{env} = |\tau_1| \times ... \times |\tau_n|\end{array}}{\begin{array}{l}\Delta; \Gamma \vdash_S \texttt{fun}\ f(x : \tau).e : \tau \rightarrow \tau' \rightsquigarrow \texttt{pack}[\tau_{env}, \\ \quad \langle(\widehat{\texttt{fun}}\ f(y : |\tau| \times \tau_{env}).[\texttt{pack}[\tau_{env}, \langle f, \pi_2 y\rangle]\ \texttt{as}\ |\tau \rightarrow \tau'|/f] \\ \quad\quad [\pi_1 y/x][\pi_1\pi_2 y/x_1]...[\pi_1\pi_2...\pi_2 y/x_{n-1}][\pi_2...\pi_2 y/x_n]\bar{e} \\ \quad), \langle x_1, \langle ... \langle x_{n-1}, x_n\rangle ...\rangle\rangle\rangle]\ \texttt{as}\ |\tau \rightarrow \tau'|\end{array}}\ Rfun$$

$$\frac{\Delta; \Gamma \vdash_S e_1 : \tau \rightarrow \tau' \rightsquigarrow \bar{e_1} \quad \Delta; \Gamma \vdash_S e_2 : \tau \rightsquigarrow \bar{e_2}}{\Delta; \Gamma \vdash_S e_1\ e_2 : \tau' \rightsquigarrow \texttt{unpack}[\alpha, x] = \bar{e_1}\ \texttt{in}(\pi_1 x)\hat{\ }\langle \bar{e_2}, \pi_2 x\rangle}\ Rapp$$

Figure 4: The closure conversion translation rules for functions and applications

4.1 Fix vs. Recursive Functions

One relevant note to discuss is the fact that we used inherently recursive functions instead of the more general fix operator. We also didn't use recursive types, as in other similar works [3], though that was a result of the added complexity that they bring. The reason for not using `fix` is because of a problem that arises from performing closure conversion on terms that contain `fix` . Suppose in our language we instead had simple lambdas, and had `fix` with the static and dynamic rules found in Figure 5.

$$\frac{\Delta; \Gamma, x : \tau \vdash e : \tau}{\Delta; \Gamma \vdash \texttt{fix } (x : \tau).e : \tau} \; Tfix$$

$$\frac{}{\texttt{fix } (x : \tau).e \mapsto [\texttt{fix } (x : \tau).e/x]e} \; Efix$$

Figure 5: Static and dynamic rules for fix

Now we can construct the term $\texttt{fix } (f : \texttt{int} \to \texttt{int}).\lambda x : \texttt{int}.\texttt{ifz}(x, 0, f\ (x-1))$. This doesn't do anything interesting, but it demonstrates the problem with fix and closure conversion. After closure conversion, this translates to

$$\texttt{fix } (f : \texttt{int} \to \texttt{int}).\,\texttt{pack}[\texttt{int} \to \texttt{int}, \langle \widehat{\lambda} y : \texttt{int} \times (\texttt{int} \to \texttt{int}).\texttt{ifz}(\pi_1 y, 0, E), f \rangle$$

Where $E = \texttt{unpack}[\alpha, z] = \pi_2 y\, \texttt{in}(\pi_1 z)\hat{}\langle \pi_1 y - 1, \pi_2 z \rangle$. Define the above to be F. Then this will step to

$$\texttt{pack}[\texttt{int} \to \texttt{int}, \langle \widehat{\lambda} y : \texttt{int} \times (\texttt{int} \to \texttt{int}).\texttt{ifz}(\pi_1 y, 0, E), F \rangle$$

This is because the whole term gets substituted in for f due to the definition of fix . Call the above F'. However, for this to be a value, we need the inner term to also be a value (that is, the term $\langle \widehat{\lambda} y : \texttt{int} \times (\texttt{int} \to \texttt{int}).\texttt{ifz}(\pi_1 y, 0, E), F \rangle$). But we just said that F steps to F', so the second term in the pair can make a step to F'. But this can also step, so this goes forever and we never reach a value. This is the problem with fix , as it does not work with closure conversion, which is why recursive functions were used instead.

5 Contextual Equivalence

The idea behind contextual equivalence is that terms are equivalent if, when both are put into any program context (a program with a hole in it), they will behave the same (that is, have the same terminating behavior). This is a practical concept of equivalence because it basically states that these two programs will always do the same thing no matter how they are used, so for practical purposes they might as well be the same.

More formally, we say that that two expressions in the combined language, $\Delta; \Gamma \vdash e : \tau$ and $\Delta; \Gamma \vdash e' : \tau$, are contextually equivalent, which we write $\Delta; \Gamma \vdash e \cong e' : \tau$, if and only if for every program context $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$, $\mathcal{C}\{e\} \simeq \mathcal{C}\{e'\}$, where $\simeq$ is Kleene equivalence, which basically says that $\mathcal{C}\{e\}$ and $\mathcal{C}\{e'\}$ have the same terminating behavior. The various different contexts are outlined in appendix D.

Unfortunately contextual equivalence is not very useful for proving properties since it isn't really possible to check all possible contexts and verify that they behave the same. So we developed a logical equivalence, similar to those found in references [1][2][4]. We then prove that this logical equivalence coincides with contextual equivalence, so that we can use them interchangebly. This is important for many of the proofs that we need. The full development of this logical relation, along with the many associated proofs, can be found in appendix E.

What we would ideally like is that, given that $\Delta; \Gamma \vdash_S e \rightsquigarrow \bar{e} : \tau$, that is that e translates to $\bar{e}$, we want to be able to say that e and $\bar{e}$ are contextually equivalent. But this isn't possible, as we have that $\Delta; \Gamma \vdash e : \tau$ and $\Delta; |\Gamma| \vdash \bar{e} : |\tau|$, and it is quite probable that τ and $|\tau|$ are not the same (which is the case if τ includes any arrow type). Thus we have no type at which we can say the two are contextually equivalent. To resolve this, we developed two functions in the combined language: $\texttt{over}_\tau$ and $\texttt{back}_\tau$, the definitions for which appear in Figure 6. Note that we use lambdas (λ) in the definitions for conciseness, they merely represent recursive functions that do not use their function variable.

$$
\begin{aligned}
\texttt{over}_\alpha &= \lambda x : \alpha . x \\
\texttt{over}_{\texttt{unit}} &= \lambda x : \texttt{unit}. x \\
\texttt{over}_{\texttt{int}} &= \lambda x : \texttt{int}. x \\
\texttt{over}_{\tau_1 \times \tau_2} &= \lambda x : \tau_1 \times \tau_2 . \langle \texttt{over}_{\tau_1} \pi_1 x, \texttt{over}_{\tau_2} \pi_2 x \rangle \\
\texttt{over}_{\tau_1 \to \tau_2} &= \lambda f : \tau_1 \to \tau_2 . \, \texttt{pack}[\tau_1 \to \tau_2, \langle \widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2). \\
&\qquad \texttt{over}_{\tau_2}((\pi_2 y) \ (\texttt{back}_{\tau_1} \pi_1 y)), f \rangle] \, \texttt{as} \, |\tau_1 \to \tau_2| \\
\texttt{over}_{\forall \alpha . \tau} &= \lambda x : (\forall \alpha . \tau) . \Lambda \alpha . (\texttt{over}_\tau (x[\alpha])) \\
\texttt{over}_{\exists \alpha . \tau} &= \lambda x : (\exists \alpha . \tau) . \, \texttt{unpack}[\alpha, y] = x \, \texttt{in} (\texttt{pack}[\alpha, \texttt{over}_\tau (y)] \, \texttt{as} \, |\exists \alpha . \tau|) \\
\texttt{back}_\alpha &= \lambda x : \alpha . x \\
\texttt{back}_{\texttt{unit}} &= \lambda x : \texttt{unit}. x \\
\texttt{back}_{\texttt{int}} &= \lambda x : \texttt{int}. x \\
\texttt{back}_{\tau_1 \times \tau_2} &= \lambda x : |\tau_1 \times \tau_2| . \langle \texttt{back}_{\tau_1} \pi_1 x, \texttt{back}_{\tau_2} \pi_2 x \rangle \\
\texttt{back}_{\tau_1 \to \tau_2} &= \lambda f : |\tau_1 \to \tau_2| . \lambda y : \tau_1 . \, \texttt{unpack}[\alpha, g] = f \, \texttt{in} \, \texttt{back}_{\tau_2} ((\pi_1 g) \hat{\ } \langle \texttt{over}_{\tau_1} y, \pi_2 g \rangle) \\
\texttt{back}_{\forall \alpha . \tau} &= \lambda x : |\forall \alpha . \tau| . \Lambda \alpha . (\texttt{back}_\tau (x[\alpha])) \\
\texttt{back}_{\exists \alpha . \tau} &= \lambda x : |\exists \alpha . \tau| . \, \texttt{unpack}[\alpha, y] = x \, \texttt{in} (\texttt{pack}[\alpha, \texttt{back}_\tau (y)] \, \texttt{as} \, \exists \alpha . \tau)
\end{aligned}
$$

Figure 6: The definitions of $\texttt{over}_\tau$ and $\texttt{back}_\tau$

Also note that as a shorthand, we define $[\texttt{over}_\Gamma/\Gamma] = [\texttt{over}_{\tau_1}(x_1)/x_1]...[\texttt{over}_{\tau_n}(x_n)/x_n]$ for $\Gamma = x_1 : \tau_1, ..., x_n : \tau_n$. A similar definition applies to $\texttt{back}_\Gamma$.

By the definitions, it is clear that both functions are in the combined language, and have the types $\Delta; \Gamma \vdash \texttt{over}_\tau : \tau \to |\tau|$ and $\Delta; \Gamma \vdash \texttt{back}_\tau : |\tau| \to \tau$. From this it is clear their intent, as $\texttt{over}_\tau$ takes terms of type τ in the source language and converts them to terms of type $|\tau|$ in the target language (and $\texttt{back}_\tau$ does the opposite). Of course even though it may be that $\Delta; \Gamma \vdash_C \texttt{over}_\tau(e) : |\tau|$, this does not imply that $\Delta; \Gamma \vdash_T \texttt{over}_\tau(e) : |\tau|$, as e may contain terms not in the target language (and so might $\texttt{over}_\tau$, in fact). Thus these functions purpose is only for reasoning in the combined language. By looking at their definitions, it is easy to tell that neither function actually looks inside the term that it is applied to. All it does is pull apart its type at the top level to convert it to the opposite language's type, never going into the term and translating it like closure conversion does. Most cases are simple, just pulling apart then repackaging the term as the correct type. However the function cases, logically, are more complicated due to the translation being nontrivial. The function $\texttt{over}_{\tau \to \tau'}$ essentially creates an empty closure and wraps the function up so it looks like the right existential (and also converts the new argument back). Opposingly, the function $\texttt{back}_{\tau \to \tau'}$ takes an existential of the correct type and pulls out the closure and uses it to call the function with the converted argument.

The most important result of this is that we now have a way to look at compiled code as if it were really source code, and vice versa. Now we can do what we originally wanted to, that source code and its compiled code are contextually equivalent (or at least, almost). It is also necessary that these two functions are inverses of one another, that is that

$$\texttt{over}_\tau \circ \texttt{back}_\tau = id = \texttt{back}_\tau \circ \texttt{over}_\tau$$

The proof of this, along with the many other relevant proofs, can be found in appendix E. Using this and our logical equivalence we can show that if $\Delta; \Gamma \vdash_S e \rightsquigarrow \bar{e} : \tau$, then $\Delta; \Gamma \vdash e \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e}) : \tau$. By the above this also means that $\Delta; |\Gamma| \vdash \texttt{over}_\tau([\texttt{back}_\Gamma/\Gamma]e) \cong \bar{e} : |\tau|$. Since $\texttt{over}_\tau$ and $\texttt{back}_\tau$ don't actually effect the behavior of the code, these imply that the source code and the compiled code do in fact behave the same.

There is one particularly interesting point that we will note about the proof of the above theorem. There are two essentially "dual" lemmas regarding polymorphism that are the key to doing the proof. These lemmas related two slightly different uses of each of the $\texttt{over}_\tau$ and $\texttt{back}_\tau$ in our logical equivalence. The parametricity of the logical equivalence was necessary here, as it allowed us to use exactly the right relations to make the lemmas hold. One of the lemmas was necessary for proving the polymorphic application case in the main theorem, and the other lemma was necessary for proving the existential pack case. Unfortunately fully understanding the lemmas involves understanding the logical equivalence, but we will give a basic idea of the lemmas here. To see the lemmas in their entirety, look at sections 11.3 and 11.4 in appendix E.

To help understand the lemmas, we will focus specifically on one part of one of them. What this does is effectively relate $[\tau'/\alpha]\texttt{back}_\tau$ to $\texttt{back}_{[\tau'/\alpha]\tau}$ at the type $|\tau| \to [\tau'/\alpha]\tau$ in our logical equivalence. Looking at these, $[\tau'/\alpha]\texttt{back}_\tau$ converts the type of what it is called on up to τ, then leaves the α alone. On the other hand, $\texttt{back}_{[\tau'/\alpha]\tau}$ converts everything, including the α type variable. This is basically saying that, under the right relation, we can pull the type substitution out of the `back` function to the top level, saying that converting the α part of the argument is equivalent to not converting it. The other parts of the lemmas say similar things, just with different configurations of `over` and `back`. These lemmas are necessary to complete the proof cases related to polymorphic and existential types.

5.1 Erasure

However, we still have our final theorem to justify. To do this we need abstraction between our source and combined languages, as well as our combined and target languages. What this means is that, we want to be able to say that if two terms are equivalent in the source, then they are equivalent in the combined language, and similarly that if two terms are equivalent in the combined language, then they are equivalent in the target. This will let us prove the final theorem. Going from the combined language to target is simple, as the language gets smaller. However, going from the source to the combined language is a little more subtle. The reason it isn't obvious is because the combined language includes something (specifically closed recursive functions) that does not appear in the source. It is still simple, as logically the closed recursive functions don't provide any functionality that normal recursive functions don't already provide, so we can prove it with another translation, which we will call erasure. The actual translation is simple, so we won't say more about it here other than that all it does is convert closed recursive functions to normal recursive functions. For the full translation, see appendix E.

As for what we do with erasure, as mentioned above this lets us say that terms equivalent in the source language are equivalent in the combined language. Thus if we assume that terms e_1 and e_2 are contextually equivalent in the source, we know that they are contextually equivalent in the combined language. But by our earlier theorem we know that each of these is contextually equivalent to the back of their compiled selves, that is $\Delta; \Gamma \vdash e_1 \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e_1}) : \tau$ and $\Delta; \Gamma \vdash e_2 \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e_2}) : \tau$. Transitivity then gets us that $\Delta; \Gamma \vdash \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e_1}) \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e_2}) : \tau$, from which some simple reasoning using the fact that $\texttt{over}_\tau$ and $\texttt{back}_\tau$ are inverses, we can get that $\bar{e_1}$ is contextually equivalent to $\bar{e_2}$ This proves our desired final theorem, and proves that our compiler preserves equivalence.

6 Comparison with Prior Work

As mentioned previously, our work is an attempt to improve upon the work done by James T. Perconti and Amal Ahmed [3]. While their work spanned two compilation phases, we can only compare our work to the first, the closure conversion phase, since that is what our work focuses on.

The main difference is that in their work, while they do merge the source and target languages together, they still keep them separate and only go from one to another using boundaries, which are additional terms that can put a term from one language into a term for another language. In some sense, these perform a similar operation to our $\texttt{over}_\tau$ and $\texttt{back}_\tau$ functions, however $\texttt{over}_\tau$ and $\texttt{back}_\tau$ are written as functions in the combined language, so they have the advantage of not being hardwired into the language itself. In their paper they also have to handle a few special cases for type variables from one language occurring inside components of another language, which they solve using suspended type variables and lump types. Our method has no need for either of these, as in the combined language, type variables are not distinguished between the two languages. That is, a type variable in a source term and a type variable in a target term look the same in the combined language, as they are just normal type variables there. This is all due to the fact that there are no boundaries between the language. Since the two languages are completely combined, they share type variables, along with most other things.

This makes the actual reasoning about the compiler significantly simpler, as there isn't a lot of extra conversion that is taking place besides the main compilation. While our $\texttt{over}_\tau$ and $\texttt{back}_\tau$ functions do a little work, they are really just there to make types line up properly, not act as a connection between the languages that is built into the language. This is our main contribution, the improvement over prior work.

7 Future Work

We have already mentioned how our method is meant to be an improvement on the work by Perconti and Ahmed [3]. However, we have only improved the first half of the work that they did, which was the compilation phase of closure conversion. Their work also extended into an allocation phase of a compiler, going between three languages total. In the future, work could be done to apply our methodology to the allocation phase of a compiler and see if it can be made to work for that as well. It could even potentially be extended to other stages of compilation. The main difficulty will be combining the languages, as other compiler phases have a much greater difference between the source and target languages. Closure conversion was a good phase to start with to see if it was possible, though due to only having one primary difference between languages it made the combination of the languages easier to reason about.

There is certainly a lot of potential for this method to be applied to other cases, since the general approach used is not specific to the language used. It will, however, likely be much more difficult as the combined language grows in complexity.

References

[1] Karl Crary and Bob Harper. Syntactic logical relations for polymorphic and recursive types. *Electronic Notes in Theoretical Computer Science*, 172:259–299, 2007.

[2] Bob Harper. *Practical Foundations for Programming Languages*, chapter 49. Cambridge University Press, 2012.

[3] James T. Perconti and Amal Ahmed. Verifying an open compiler using multi-language semantics. *Proceedings of the 23rd European Symposium on Programming*, 2014.

[4] Benjamin Pierce, editor. *Advanced Topics in Types and Programming Languages*, chapter 7. MIT Press, 2004.

A Combined Language Statics

A.1 Types

$$\frac{}{\Delta \vdash \texttt{unit}\,\texttt{type}}\;Dunit \qquad \frac{}{\Delta \vdash \texttt{int}\,\texttt{type}}\;Dint$$

$$\frac{}{\Delta, \alpha \vdash \alpha\,\texttt{type}}\;Dvar \qquad \frac{\Delta \vdash \tau_1\,\texttt{type} \quad \Delta \vdash \tau_2\,\texttt{type}}{\Delta \vdash \tau_1 \times \tau_2\,\texttt{type}}\;Dpair$$

$$\frac{\Delta \vdash \tau_1\,\texttt{type} \quad \Delta \vdash \tau_2\,\texttt{type}}{\Delta \vdash \tau_1 \to \tau_2\,\texttt{type}}\;Dfun \qquad \frac{\Delta \vdash \tau_1\,\texttt{type} \quad \Delta \vdash \tau_2\,\texttt{type}}{\Delta \vdash \tau_1 \Rightarrow \tau_2\,\texttt{type}}\;Dccfun$$

$$\frac{\Delta, \alpha \vdash \tau\,\texttt{type}}{\Delta \vdash \forall \alpha.\tau\,\texttt{type}}\;Dforall \qquad \frac{\Delta, \alpha \vdash \tau\,\texttt{type}}{\Delta \vdash \exists \alpha.\tau\,\texttt{type}}\;Dexists$$

A.2 Terms

$$\frac{}{\Delta; \Gamma \vdash \texttt{()} : \texttt{unit}}\;Tunit \qquad \frac{\Gamma(x) = \tau}{\Delta; \Gamma \vdash x : \tau}\;Tvar$$

$$\frac{}{\Delta; \Gamma \vdash n : \texttt{int}}\;Tint \qquad \frac{\Delta; \Gamma \vdash e_1 : \texttt{int} \quad \Delta; \Gamma \vdash e_2 : \texttt{int}}{\Delta; \Gamma \vdash e_1\,\texttt{p}\,e_2 : \texttt{int}}\;Tintop$$

$$\frac{\Delta; \Gamma \vdash e_1 : \texttt{int} \quad \Delta; \Gamma \vdash e_2 : \tau \quad \Delta; \Gamma \vdash e_3 : \tau}{\Delta; \Gamma \vdash \texttt{ifz}(e_1, e_2, e_3) : \tau}\;Tifz$$

$$\frac{\Delta \vdash \tau\,\texttt{type} \quad \Delta; \Gamma, f : \tau \to \tau', x : \tau \vdash e : \tau'}{\Delta; \Gamma \vdash \texttt{fun}\ f(x : \tau).e : \tau \to \tau'}\;Tfun \qquad \frac{\Delta; \Gamma \vdash e_1 : \tau \to \tau' \quad \Delta; \Gamma \vdash e_2 : \tau}{\Delta; \Gamma \vdash e_1\ e_2 : \tau'}\;Tapp$$

$$\frac{\Delta \vdash \tau\,\texttt{type} \quad \Delta; f : \tau \to \tau', x : \tau \vdash e : \tau'}{\Delta; \Gamma \vdash \widehat{\texttt{fun}}\ f(x : \tau).e : \tau \Rightarrow \tau'}\;Tccfun \qquad \frac{\Delta; \Gamma \vdash e_1 : \tau \Rightarrow \tau' \quad \Delta; \Gamma \vdash e_2 : \tau}{\Delta; \Gamma \vdash e_1\hat{}\,e_2 : \tau'}\;Tccapp$$

$$\frac{\Delta; \Gamma \vdash e_1 : \tau_1 \quad \Delta; \Gamma \vdash e_2 : \tau_2}{\Delta; \Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2}\;Tpair \qquad \frac{\Delta; \Gamma \vdash e : \tau_1 \times \tau_2 \quad i \in \{1, 2\}}{\Delta; \Gamma \vdash \pi_i e : \tau_i}\;Tproj$$

$$\frac{\Delta, \alpha; \Gamma \vdash e : \tau}{\Delta; \Gamma \vdash \Lambda \alpha.e : \forall \alpha.\tau}\;Ttlam \qquad \frac{\Delta; \Gamma \vdash e : \forall \alpha.\tau \quad \Delta; \Gamma \vdash \tau'\,\texttt{type}}{\Delta; \Gamma \vdash e[\tau'] : [\tau'/\alpha]\tau}\;Ttapp$$

$$\frac{\Delta \vdash \tau'\,\texttt{type} \quad \Delta, \alpha \vdash \tau\,\texttt{type} \quad \Delta; \Gamma \vdash e : [\tau'/\alpha]\tau}{\Delta; \Gamma \vdash \texttt{pack}[\tau', e]\,\texttt{as}\,\exists \alpha.\tau : \exists \alpha.\tau}\;Tpack$$

$$\frac{\Delta; \Gamma \vdash e_1 : \exists \alpha.\tau_1 \quad \Delta, \alpha; \Gamma, x : \tau_1 \vdash e_2 : \tau_2 \quad \Delta \vdash \tau_2\,\texttt{type}}{\Delta; \Gamma \vdash \texttt{unpack}[\alpha, x] = e_1\,\texttt{in}\,e_2 : \tau_2}\;Tunpack$$

B Combined Language Dynamics

B.1 Values

$$\frac{}{\texttt{()}\ \texttt{val}}\ Vunit \qquad \frac{}{n\ \texttt{val}}\ Vint \qquad \frac{e_1\ \texttt{val} \quad e_2\ \texttt{val}}{\langle e_1, e_2\rangle\ \texttt{val}}\ Vpair$$

$$\frac{}{\texttt{fun}\ f(x:\tau).e\ \texttt{val}}\ Vfun \qquad \frac{}{\widehat{\texttt{fun}}\ f(x:\tau).e\ \texttt{val}}\ Vccfun$$

$$\frac{}{\Lambda\alpha.e\ \texttt{val}}\ Vtlam \qquad \frac{e\ \texttt{val}}{\texttt{pack}[\tau', e]\ \texttt{as}\ \exists\alpha.\tau\ \texttt{val}}\ Vpack$$

B.2 Evaluation

$$\frac{e_1 \mapsto e_1'}{e_1\ \texttt{p}\ e_2 \mapsto e_1'\ \texttt{p}\ e_2}\ Eintop_1 \qquad \frac{e_2 \mapsto e_2'}{n_1\ \texttt{p}\ e_2 \mapsto n_1\ \texttt{p}\ e_2'}\ Eintop_2 \qquad \frac{n_1\ \texttt{p}\ n_2 = n}{n_1\ \texttt{p}\ n_2 \mapsto n}\ Eintop_3$$

$$\frac{e_1 \mapsto e_1'}{\texttt{ifz}(e_1, e_2, e_3) \mapsto \texttt{ifz}(e_1', e_2, e_3)}\ Eifz_1$$

$$\frac{n = 0}{\texttt{ifz}(n, e_2, e_3) \mapsto e_2}\ Eifz_2 \qquad \frac{n \neq 0}{\texttt{ifz}(n, e_2, e_3) \mapsto e_3}\ Eifz_3$$

$$\frac{e_1 \mapsto e_1'}{e_1\ e_2 \mapsto e_1'\ e_2}\ Eapp_1 \qquad \frac{e_2 \mapsto e_2'}{(\texttt{fun}\ f(x:\tau).e)\ e_2 \mapsto (\texttt{fun}\ f(x:\tau).e)\ e_2'}\ Eapp_2$$

$$\frac{e_2\ \texttt{val}}{(\texttt{fun}\ f(x:\tau).e)\ e_2 \mapsto [\texttt{fun}\ f(x:\tau).e/f][e_2/x]e}\ Eapp_3$$

$$\frac{e_1 \mapsto e_1'}{e_1\hat{}\,e_2 \mapsto e_1'\hat{}\,e_2}\ Eccapp_1 \qquad \frac{e_2 \mapsto e_2'}{(\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{}\,e_2 \mapsto (\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{}\,e_2'}\ Eccapp_2$$

$$\frac{e_2\ \texttt{val}}{(\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{}\,e_2 \mapsto [\widehat{\texttt{fun}}\ f(x:\tau).e/f][e_2/x]e}\ Eccapp_3$$

$$\frac{e_1 \mapsto e_1'}{\langle e_1, e_2\rangle \mapsto \langle e_1', e_2\rangle}\ Epair_1 \qquad \frac{e_1\ \texttt{val} \quad e_2 \mapsto e_2'}{\langle e_1, e_2\rangle \mapsto \langle e_1, e_2'\rangle}\ Epair_2$$

$$\frac{e \mapsto e'}{\pi_i e \mapsto \pi_i e'}\ Eproj_1 \qquad \frac{i \in \{1,2\} \quad e_1\ \texttt{val} \quad e_2\ \texttt{val}}{\pi_i\langle e_1, e_2\rangle \mapsto e_i}\ Eproj_2$$

$$\frac{e \mapsto e'}{e[\tau] \mapsto e'[\tau]}\ Etapp_1 \qquad \frac{}{(\Lambda\alpha.e)[\tau] \mapsto [\tau/\alpha]e}\ Etapp_2$$

$$\frac{e \mapsto e'}{\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau \mapsto \texttt{pack}[\tau', e']\,\texttt{as}\,\exists\alpha.\tau}\ \textit{Epack}$$

$$\frac{e_1 \mapsto e_1'}{\texttt{unpack}[\alpha, x] = e_1\,\texttt{in}\,e_2 \mapsto \texttt{unpack}[\alpha, x] = e_1'\,\texttt{in}\,e_2}\ \textit{Eunpack}_1$$

$$\frac{e\,\texttt{val}}{\texttt{unpack}[\alpha, x] = (\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau)\,\texttt{in}\,e_2 \mapsto [\tau'/\alpha][e/x]e_2}\ \textit{Eunpack}_2$$

C Closure Conversion

$$\frac{}{\Delta; \Gamma \vdash_S \texttt{()} : \texttt{unit} \leadsto \texttt{()}}\ \textit{Runit} \qquad \frac{\Gamma(x) = \tau}{\Delta; \Gamma \vdash_S x : \tau \leadsto x}\ \textit{Rvar}$$

$$\frac{}{\Delta; \Gamma \vdash_S n : \texttt{int} \leadsto n}\ \textit{Rint} \qquad \frac{\Delta; \Gamma \vdash_S e_1 : \texttt{int} \leadsto \bar{e_1} \quad \Delta; \Gamma \vdash_S e_2 : \texttt{int} \leadsto \bar{e_2}}{\Delta; \Gamma \vdash_S e_1\,\texttt{p}\,e_2 : \texttt{int} \leadsto \bar{e_1}\,\texttt{p}\,\bar{e_2}}\ \textit{Rintop}$$

$$\frac{\Delta; \Gamma \vdash_S e_1 : \texttt{int} \leadsto \bar{e_1} \quad \Delta; \Gamma \vdash_S e_2 : \tau \leadsto \bar{e_2} \quad \Delta; \Gamma \vdash_S e_3 : \tau \leadsto \bar{e_3}}{\Delta; \Gamma \vdash_S \texttt{ifz}(e_1, e_2, e_3) : \tau \leadsto \texttt{ifz}(\bar{e_1}, \bar{e_2}, \bar{e_3})}\ \textit{Rifz}$$

$$\frac{\Delta; \Gamma \vdash_S e_1 : \tau_1 \leadsto \bar{e_1} \quad \Delta; \Gamma \vdash_S e_2 : \tau_2 \leadsto \bar{e_2}}{\Delta; \Gamma \vdash_S \langle e_1, e_2 \rangle : \tau_1 \times \tau_2 \leadsto \langle \bar{e_1}, \bar{e_2} \rangle}\ \textit{Rpair}$$

$$\frac{\Delta; \Gamma \vdash_S e : \tau_1 \times \tau_2 \leadsto \bar{e} \quad i \in \{1, 2\}}{\Delta; \Gamma \vdash_S \pi_i e : \tau_i \leadsto \pi_i \bar{e}}\ \textit{Rproj}$$

$$\frac{\Delta, \alpha; \Gamma \vdash_S e : \tau \leadsto \bar{e}}{\Delta; \Gamma \vdash_S \Lambda\alpha.e : \forall\alpha.\tau \leadsto \Lambda\alpha.\bar{e}}\ \textit{Rtlam} \qquad \frac{\Delta; \Gamma \vdash_S e : \forall\alpha.\tau \leadsto \bar{e} \quad \Delta \vdash_S \tau'\,\texttt{type}}{\Delta; \Gamma \vdash_S e[\tau'] : [\tau'/\alpha]\tau \leadsto \bar{e}[|\tau'|]}\ \textit{Rtapp}$$

$$\frac{\Delta \vdash_S \tau'\,\texttt{type} \quad \Delta, \alpha \vdash_S \tau\,\texttt{type} \quad \Delta; \Gamma \vdash_S e : [\tau'/\alpha]\tau \leadsto \bar{e}}{\Delta; \Gamma \vdash_S \texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau : \exists\alpha.\tau \leadsto \texttt{pack}[|\tau'|, \bar{e}]\,\texttt{as}\,\exists\alpha.|\tau|}\ \textit{Rpack}$$

$$\frac{\Delta; \Gamma \vdash_S e_1 : \exists\alpha.\tau_1 \leadsto \bar{e_1} \quad \Delta, \alpha; \Gamma, x : \tau_1 \vdash_S e_2 : \tau_2 \leadsto \bar{e_2} \quad \Delta \vdash_S \tau_2\,\texttt{type}}{\Delta; \Gamma \vdash_S \texttt{unpack}[\alpha, x] = e_1\,\texttt{in}\,e_2 : \tau_2 \leadsto \texttt{unpack}[\alpha, x] = \bar{e_1}\,\texttt{in}\,\bar{e_2}}\ \textit{Runpack}$$

$$\frac{\begin{array}{c}\Gamma = x_1 : \tau_1, ..., x_n : \tau_n \qquad \Delta \vdash_S \tau\,\texttt{type} \\ \Delta; \Gamma, x : \tau, f : \tau \to \tau' \vdash_S e : \tau' \leadsto \bar{e} \quad \tau_{env} = |\tau_1| \times ... \times |\tau_n|\end{array}}{\begin{array}{l}\Delta; \Gamma \vdash_S \texttt{fun}\ f(x : \tau).e : \tau \to \tau' \leadsto \texttt{pack}[\tau_{env}, \\ \quad \langle (\widehat{\texttt{fun}}\ f(y : |\tau| \times \tau_{env}).[\texttt{pack}[\tau_{env}, \langle f, \pi_2 y \rangle]\,\texttt{as}\,|\tau \to \tau'|/f] \\ \qquad [\pi_1 y/x][\pi_1\pi_2 y/x_1]...[\pi_1\pi_2...\pi_2 y/x_{n-1}][\pi_2...\pi_2 y/x_n]\bar{e} \\ \quad), \langle x_1, \langle ... \langle x_{n-1}, x_n \rangle ... \rangle \rangle \rangle]\,\texttt{as}\,|\tau \to \tau'|\end{array}}\ \textit{Rfun}$$

$$\frac{\Delta;\Gamma \vdash_S e_1 : \tau \to \tau' \rightsquigarrow \bar{e_1} \quad \Delta;\Gamma \vdash_S e_2 : \tau \rightsquigarrow \bar{e_2}}{\Delta;\Gamma \vdash_S e_1\ e_2 : \tau' \rightsquigarrow \texttt{unpack}[\alpha, x] = \bar{e_1}\ \texttt{in}(\pi_1 x)\hat{\ }\langle \bar{e_2}, \pi_2 x\rangle}\ \mathit{Rapp}$$

D Contexts

$$\frac{}{\circ : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta;\Gamma \triangleright \tau)}\ \mathit{Cid}$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \texttt{int}) \quad \Delta';\Gamma' \vdash e : \texttt{int}}{\mathcal{C}\,\texttt{p}\,e : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \texttt{int})}\ \mathit{Cintop}_1$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \texttt{int}) \quad \Delta';\Gamma' \vdash e : \texttt{int}}{e\,\texttt{p}\,\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \texttt{int})}\ \mathit{Cintop}_2$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \texttt{int}) \quad \Delta';\Gamma' \vdash e_2 : \tau' \quad \Delta';\Gamma' \vdash e_3 : \tau'}{\texttt{ifz}(\mathcal{C}, e_2, e_3) : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')}\ \mathit{Cifz}_1$$

$$\frac{\Delta';\Gamma' \vdash e_1 : \texttt{int} \quad \mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau') \quad \Delta';\Gamma' \vdash e_3 : \tau'}{\texttt{ifz}(e_1, \mathcal{C}, e_3) : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')}\ \mathit{Cifz}_2$$

$$\frac{\Delta';\Gamma' \vdash e_1 : \texttt{int} \quad \mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau') \quad \Delta';\Gamma' \vdash e_2 : \tau'}{\texttt{ifz}(e_1, e_2, \mathcal{C}) : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')}\ \mathit{Cifz}_3$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma', f : \tau_1 \to \tau_2, x : \tau_1 \triangleright \tau_2) \quad \Delta' \vdash \tau_1\ \texttt{type}}{\texttt{fun}\ f(x : \tau_1).\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_1 \to \tau_2)}\ \mathit{Cfun}$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta; f : \tau_1 \to \tau_2, x : \tau_1 \triangleright \tau_2) \quad \Delta' \vdash \tau_1\ \texttt{type}}{\widehat{\texttt{fun}}\ f(x : \tau_1).\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_1 \Rightarrow \tau_2)}\ \mathit{Cccfun}$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_1 \to \tau_2) \quad \Delta';\Gamma' \vdash e : \tau_1}{\mathcal{C}\ e : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_2)}\ \mathit{Capp}_1$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_1) \quad \Delta';\Gamma' \vdash e : \tau_1 \to \tau_2}{e\ \mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_2)}\ \mathit{Capp}_2$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_1 \Rightarrow \tau_2) \quad \Delta';\Gamma' \vdash e : \tau_1}{\mathcal{C}\hat{\ }e : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_2)}\ \mathit{Cccapp}_1$$

$$\frac{\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_1) \quad \Delta';\Gamma' \vdash e : \tau_1 \Rightarrow \tau_2}{e\hat{\ }\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau_2)}\ \mathit{Cccapp}_2$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1) \quad \Delta'; \Gamma' \vdash e : \tau_2}{\langle \mathcal{C}, e \rangle : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \times \tau_2)} \; Cpair_1$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2) \quad \Delta'; \Gamma' \vdash e : \tau_1}{\langle e, \mathcal{C} \rangle : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \times \tau_2)} \; Cpair_2$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \times \tau_2) \quad i \in \{1, 2\}}{\pi_i \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_i)} \; Cproj$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta', \alpha; \Gamma' \triangleright \tau')}{\Lambda \alpha . \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \forall \alpha . \tau')} \; Ctlam$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \forall \alpha . \tau') \quad \Delta' \vdash \tau'' \; \texttt{type}}{\mathcal{C}[\tau''] : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright [\tau''/\alpha]\tau')} \; Ctapp$$

$$\frac{\Delta' \vdash \tau'' \; \texttt{type} \quad \Delta', \alpha \vdash \tau' \; \texttt{type} \quad \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright [\tau''/\alpha]\tau')}{\texttt{pack}[\tau'', \mathcal{C}] \; \texttt{as} \; \exists \alpha . \tau' : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \exists \alpha . \tau')} \; Cpack$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \exists \alpha . \tau_1) \quad \Delta', \alpha; \Gamma', x : \tau_1 \vdash e : \tau_2 \quad \Delta' \vdash \tau_2 \; \texttt{type}}{\texttt{unpack}[\alpha, x] = \mathcal{C} \; \texttt{in} \; e : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2)} \; Cunpack_1$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta', \alpha; \Gamma', x : \tau_1 \triangleright \tau_2) \quad \Delta'; \Gamma' \vdash e : \exists \alpha . \tau_1 \quad \Delta' \vdash \tau_2 \; \texttt{type}}{\texttt{unpack}[\alpha, x] = e \; \texttt{in} \; \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2)} \; Cunpack_2$$

E Proofs

The following is a full account of all proofs involved in the above paper. Some of the above information will be repeated, as they are mentioned where they first appear in the proofs. The proofs read in order, so any lemmas/theorems used in a proof will have been proven before it.

Contents

1 Languages **3**
1.1 Source Language . . . 3
1.2 Target Language . . . 3
1.3 Combined Language . . . 3

2 Combined Statics **4**
2.1 Types . . . 4
2.2 Terms . . . 4

3 Combined Dynamics **6**
3.1 Values . . . 6
3.2 Evaluation . . . 6
3.3 Termination . . . 7

4 Contexts **8**
4.1 Context Composition . . . 10

5 Contextual Equivalence **13**
5.1 Definitions . . . 13
5.2 Substitutivity . . . 15

6 Compactness **20**
6.1 Simulation . . . 22
6.2 Compactness . . . 33

7 Relations **34**
7.1 ST Closure . . . 34
7.2 Admissibility . . . 36

8 Logical Equivalence **37**
8.1 Compositionality . . . 40

9 Logical and Contextual Equivalence Coincide **43**
9.1 Reflexivity . . . 43
9.2 Congruence . . . 51
9.3 Respect for Contextual Equivalence . . . 51
9.4 Logical Equivalence implies Contextual Equivalence . . . 52
9.5 Contextual Equivalence implies Logical Equivalence . . . 52

10 Closure Conversion **53**

10.1 Translation 53

10.2 Fix vs. Recursive Functions 54

11 Language Conversion **55**

11.1 Over and Back 55

11.2 Inverses 55

11.3 The Back Relation 63

11.4 The Over Relation 73

11.5 Translation Equivalence 83

12 Erasure **94**

12.1 Static Erasure 94

12.2 Dynamic Erasure 97

13 Equivalence Preservation **101**

1 Languages

1.1 Source Language

$$
\begin{aligned}
\tau &::= \alpha \mid \texttt{unit} \mid \texttt{int} \mid \tau \times \tau \mid \tau \to \tau \mid \forall\alpha.\tau \mid \exists\alpha.\tau \\
e &::= \texttt{()} \mid n \mid e \,\texttt{p}\, e \mid \texttt{ifz}(e, e, e) \mid x \mid \langle e, e\rangle \mid \pi_i e \mid \texttt{fun}\ f(x:\tau).e \mid e\ e \mid \Lambda\alpha.e \mid e[\tau] \\
&\quad \mid \texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau \mid \texttt{unpack}[\alpha, x] = e\,\texttt{in}\,e \\
\texttt{p} &::= + \mid - \mid * \\
\Gamma &::= \cdot \mid \Gamma, x:\tau \\
\Delta &::= \cdot \mid \Delta, \alpha
\end{aligned}
$$

1.2 Target Language

$$
\begin{aligned}
\tau &::= \alpha \mid \texttt{unit} \mid \texttt{int} \mid \tau \times \tau \mid \tau \Rightarrow \tau \mid \forall\alpha.\tau \mid \exists\alpha.\tau \\
e &::= \texttt{()} \mid n \mid e \,\texttt{p}\, e \mid \texttt{ifz}(e, e, e) \mid x \mid \langle e, e\rangle \mid \pi_i e \mid \widehat{\texttt{fun}}\ f(x:\tau).e \mid e\,\hat{}\,e \mid \Lambda\alpha.e \mid e[\tau] \\
&\quad \mid \texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau \mid \texttt{unpack}[\alpha, x] = e\,\texttt{in}\,e \\
\texttt{p} &::= + \mid - \mid * \\
\Gamma &::= \cdot \mid \Gamma, x:\tau \\
\Delta &::= \cdot \mid \Delta, \alpha
\end{aligned}
$$

As a shorthand, we write $\lambda x : \tau.e$ to stand for $\texttt{fun}\ f(x : \tau).e$ when f is not used in e (meaning that $x : \tau \vdash e : \tau'$, assuming otherwise closed terms). Similarly, we write $\widehat{\lambda} x : \tau.e$ to stand for $\widehat{\texttt{fun}}\ f(x : \tau).e$ when f is not used in e (meaning that $x : \tau \vdash e : \tau'$, assuming otherwise closed terms).

1.3 Combined Language

$$
\begin{aligned}
\tau &::= \alpha \mid \texttt{unit} \mid \texttt{int} \mid \tau \times \tau \mid \tau \to \tau \mid \tau \Rightarrow \tau \mid \forall\alpha.\tau \mid \exists\alpha.\tau \\
e &::= \texttt{()} \mid n \mid e \,\texttt{p}\, e \mid \texttt{ifz}(e, e, e) \mid x \mid \langle e, e\rangle \mid \pi_i e \mid \texttt{fun}\ f(x:\tau).e \mid e\ e \mid \widehat{\texttt{fun}}\ f(x:\tau).e \mid e\,\hat{}\,e \\
&\quad \mid \Lambda\alpha.e \mid e[\tau] \mid \texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau \mid \texttt{unpack}[\alpha, x] = e\,\texttt{in}\,e \\
\texttt{p} &::= + \mid - \mid * \\
\Gamma &::= \cdot \mid \Gamma, x:\tau \\
\Delta &::= \cdot \mid \Delta, \alpha
\end{aligned}
$$

2 Combined Statics

2.1 Types

$$\frac{}{\Delta \vdash \texttt{unit}\,\texttt{type}}\ Dunit \qquad \frac{}{\Delta \vdash \texttt{int}\,\texttt{type}}\ Dint$$

$$\frac{}{\Delta, \alpha \vdash \alpha\,\texttt{type}}\ Dvar \qquad \frac{\Delta \vdash \tau_1\,\texttt{type} \quad \Delta \vdash \tau_2\,\texttt{type}}{\Delta \vdash \tau_1 \times \tau_2\,\texttt{type}}\ Dpair$$

$$\frac{\Delta \vdash \tau_1\,\texttt{type} \quad \Delta \vdash \tau_2\,\texttt{type}}{\Delta \vdash \tau_1 \to \tau_2\,\texttt{type}}\ Dfun \qquad \frac{\Delta \vdash \tau_1\,\texttt{type} \quad \Delta \vdash \tau_2\,\texttt{type}}{\Delta \vdash \tau_1 \Rightarrow \tau_2\,\texttt{type}}\ Dccfun$$

$$\frac{\Delta, \alpha \vdash \tau\,\texttt{type}}{\Delta \vdash \forall \alpha.\tau\,\texttt{type}}\ Dforall \qquad \frac{\Delta, \alpha \vdash \tau\,\texttt{type}}{\Delta \vdash \exists \alpha.\tau\,\texttt{type}}\ Dexists$$

2.2 Terms

$$\frac{}{\Delta; \Gamma \vdash \texttt{()} : \texttt{unit}}\ Tunit \qquad \frac{\Gamma(x) = \tau}{\Delta; \Gamma \vdash x : \tau}\ Tvar$$

$$\frac{}{\Delta; \Gamma \vdash n : \texttt{int}}\ Tint \qquad \frac{\Delta; \Gamma \vdash e_1 : \texttt{int} \quad \Delta; \Gamma \vdash e_2 : \texttt{int}}{\Delta; \Gamma \vdash e_1\,\texttt{p}\,e_2 : \texttt{int}}\ Tintop$$

$$\frac{\Delta; \Gamma \vdash e_1 : \texttt{int} \quad \Delta; \Gamma \vdash e_2 : \tau \quad \Delta; \Gamma \vdash e_3 : \tau}{\Delta; \Gamma \vdash \texttt{ifz}(e_1, e_2, e_3) : \tau}\ Tifz$$

$$\frac{\Delta \vdash \tau\,\texttt{type} \quad \Delta; \Gamma, f : \tau \to \tau', x : \tau \vdash e : \tau'}{\Delta; \Gamma \vdash \texttt{fun}\ f(x : \tau).e : \tau \to \tau'}\ Tfun \qquad \frac{\Delta; \Gamma \vdash e_1 : \tau \to \tau' \quad \Delta; \Gamma \vdash e_2 : \tau}{\Delta; \Gamma \vdash e_1\ e_2 : \tau'}\ Tapp$$

$$\frac{\Delta \vdash \tau\,\texttt{type} \quad \Delta; f : \tau \to \tau', x : \tau \vdash e : \tau'}{\Delta; \Gamma \vdash \widehat{\texttt{fun}}\ f(x : \tau).e : \tau \Rightarrow \tau'}\ Tccfun \qquad \frac{\Delta; \Gamma \vdash e_1 : \tau \Rightarrow \tau' \quad \Delta; \Gamma \vdash e_2 : \tau}{\Delta; \Gamma \vdash e_1 \hat{}\, e_2 : \tau'}\ Tccapp$$

$$\frac{\Delta; \Gamma \vdash e_1 : \tau_1 \quad \Delta; \Gamma \vdash e_2 : \tau_2}{\Delta; \Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2}\ Tpair \qquad \frac{\Delta; \Gamma \vdash e : \tau_1 \times \tau_2 \quad i \in \{1, 2\}}{\Delta; \Gamma \vdash \pi_i e : \tau_i}\ Tproj$$

$$\frac{\Delta, \alpha; \Gamma \vdash e : \tau}{\Delta; \Gamma \vdash \Lambda \alpha.e : \forall \alpha.\tau}\ Ttlam \qquad \frac{\Delta; \Gamma \vdash e : \forall \alpha.\tau \quad \Delta; \Gamma \vdash \tau'\,\texttt{type}}{\Delta; \Gamma \vdash e[\tau'] : [\tau'/\alpha]\tau}\ Ttapp$$

$$\frac{\Delta \vdash \tau'\,\texttt{type} \quad \Delta, \alpha \vdash \tau\,\texttt{type} \quad \Delta; \Gamma \vdash e : [\tau'/\alpha]\tau}{\Delta; \Gamma \vdash \texttt{pack}[\tau', e]\,\texttt{as}\,\exists \alpha.\tau : \exists \alpha.\tau}\ Tpack$$

$$\frac{\Delta; \Gamma \vdash e_1 : \exists \alpha.\tau_1 \quad \Delta, \alpha; \Gamma, x : \tau_1 \vdash e_2 : \tau_2 \quad \Delta \vdash \tau_2\ \texttt{type}}{\Delta; \Gamma \vdash \texttt{unpack}[\alpha, x] = e_1\ \texttt{in}\ e_2 : \tau_2}\ Tunpack$$

The statics are the same for the source language, except without rules $Dccfun$, $Tccfun$, and $Tccapp$. We also say that $\Delta; \Gamma \vdash_S e : \tau$ to indicate that e can be typed at τ in the source language.

The statics are the same for the target language, except without rules $Dfun$, $Tfun$, and $Tapp$. We also say that $\Delta; \Gamma \vdash_T e : \tau$ to indicate that e can be typed at τ in the target language.

Clearly if either $\Delta; \Gamma \vdash_S e : \tau$ or $\Delta; \Gamma \vdash_T e : \tau$ then $\Delta; \Gamma \vdash e : \tau$, in the combined language.

3 Combined Dynamics

3.1 Values

$$\frac{}{\texttt{()}\ \texttt{val}}\ Vunit \qquad \frac{}{n\ \texttt{val}}\ Vint \qquad \frac{e_1\ \texttt{val} \quad e_2\ \texttt{val}}{\langle e_1, e_2\rangle\ \texttt{val}}\ Vpair$$

$$\frac{}{\texttt{fun}\ f(x:\tau).e\ \texttt{val}}\ Vfun \qquad \frac{}{\widehat{\texttt{fun}}\ f(x:\tau).e\ \texttt{val}}\ Vccfun$$

$$\frac{}{\Lambda\alpha.e\ \texttt{val}}\ Vtlam \qquad \frac{e\ \texttt{val}}{\texttt{pack}[\tau', e]\ \texttt{as}\ \exists\alpha.\tau\ \texttt{val}}\ Vpack$$

3.2 Evaluation

$$\frac{e_1 \mapsto e_1'}{e_1\ \texttt{p}\ e_2 \mapsto e_1'\ \texttt{p}\ e_2}\ Eintop_1 \qquad \frac{e_2 \mapsto e_2'}{n_1\ \texttt{p}\ e_2 \mapsto n_1\ \texttt{p}\ e_2'}\ Eintop_2 \qquad \frac{n_1\ \texttt{p}\ n_2 = n}{n_1\ \texttt{p}\ n_2 \mapsto n}\ Eintop_3$$

$$\frac{e_1 \mapsto e_1'}{\texttt{ifz}(e_1, e_2, e_3) \mapsto \texttt{ifz}(e_1', e_2, e_3)}\ Eifz_1$$

$$\frac{n = 0}{\texttt{ifz}(n, e_2, e_3) \mapsto e_2}\ Eifz_2 \qquad \frac{n \neq 0}{\texttt{ifz}(n, e_2, e_3) \mapsto e_3}\ Eifz_3$$

$$\frac{e_1 \mapsto e_1'}{e_1\ e_2 \mapsto e_1'\ e_2}\ Eapp_1 \qquad \frac{e_2 \mapsto e_2'}{(\texttt{fun}\ f(x:\tau).e)\ e_2 \mapsto (\texttt{fun}\ f(x:\tau).e)\ e_2'}\ Eapp_2$$

$$\frac{e_2\ \texttt{val}}{(\texttt{fun}\ f(x:\tau).e)\ e_2 \mapsto [\texttt{fun}\ f(x:\tau).e/f][e_2/x]e}\ Eapp_3$$

$$\frac{e_1 \mapsto e_1'}{e_1\hat{\ }e_2 \mapsto e_1'\hat{\ }e_2}\ Eccapp_1 \qquad \frac{e_2 \mapsto e_2'}{(\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{\ }e_2 \mapsto (\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{\ }e_2'}\ Eccapp_2$$

$$\frac{e_2\ \texttt{val}}{(\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{\ }e_2 \mapsto [\widehat{\texttt{fun}}\ f(x:\tau).e/f][e_2/x]e}\ Eccapp_3$$

$$\frac{e_1 \mapsto e_1'}{\langle e_1, e_2\rangle \mapsto \langle e_1', e_2\rangle}\ Epair_1 \qquad \frac{e_1\ \texttt{val} \quad e_2 \mapsto e_2'}{\langle e_1, e_2\rangle \mapsto \langle e_1, e_2'\rangle}\ Epair_2$$

$$\frac{e \mapsto e'}{\pi_i e \mapsto \pi_i e'}\ Eproj_1 \qquad \frac{i \in \{1,2\} \quad e_1\ \texttt{val} \quad e_2\ \texttt{val}}{\pi_i\langle e_1, e_2\rangle \mapsto e_i}\ Eproj_2$$

$$\frac{e \mapsto e'}{e[\tau] \mapsto e'[\tau]}\ Etapp_1 \qquad \frac{}{(\Lambda\alpha.e)[\tau] \mapsto [\tau/\alpha]e}\ Etapp_2$$

$$\frac{e \mapsto e'}{\texttt{pack}[\tau', e]\ \texttt{as}\ \exists\alpha.\tau \mapsto \texttt{pack}[\tau', e']\ \texttt{as}\ \exists\alpha.\tau}\ \textit{Epack}$$

$$\frac{e_1 \mapsto e_1'}{\texttt{unpack}[\alpha, x] = e_1\ \texttt{in}\ e_2 \mapsto \texttt{unpack}[\alpha, x] = e_1'\ \texttt{in}\ e_2}\ \textit{Eunpack}_1$$

$$\frac{e\ \texttt{val}}{\texttt{unpack}[\alpha, x] = (\texttt{pack}[\tau', e]\ \texttt{as}\ \exists\alpha.\tau)\ \texttt{in}\ e_2 \mapsto [\tau'/\alpha][e/x]e_2}\ \textit{Eunpack}_2$$

The dynamics are the same for the source language, except without rules $Vccfun$, $Eccapp_1$, $Eccapp_2$, and $Eccapp_3$.

The dynamics are the same for the source language, except without rules $Vfun$, $Eapp_1$, $Eapp_2$, and $Eapp_3$.

3.3 Termination

Definition An expression terminates (or halts) if it steps to a value after a finite number of steps. With the introduction of recursive functions, we can now have divergent expressions. We define termination, written $e\downarrow$, as follows:

$$\frac{e\ \texttt{val}}{e\downarrow}\ Hval \qquad \frac{e \mapsto e' \quad e'\downarrow}{e\downarrow}\ Hstep$$

For simplicity, from now on consider v to be an expression e such that $e\ \texttt{val}$.

Also, for convenience we define the symbol $\perp$ to be any term that does not terminate. An example of this is

$$\perp = (\texttt{fun}\ f(x : \texttt{unit}).f\ x)\texttt{()}$$

This clearly does not terminate, as it steps to itself.

4 Contexts

$$\frac{}{o : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta; \Gamma \triangleright \tau)}\ Cid$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \texttt{int}) \quad \Delta'; \Gamma' \vdash e : \texttt{int}}{\mathcal{C}\,\texttt{p}\,e : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \texttt{int})}\ Cintop_1$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \texttt{int}) \quad \Delta'; \Gamma' \vdash e : \texttt{int}}{e\,\texttt{p}\,\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \texttt{int})}\ Cintop_2$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \texttt{int}) \quad \Delta'; \Gamma' \vdash e_2 : \tau' \quad \Delta'; \Gamma' \vdash e_3 : \tau'}{\texttt{ifz}(\mathcal{C}, e_2, e_3) : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')}\ Cifz_1$$

$$\frac{\Delta'; \Gamma' \vdash e_1 : \texttt{int} \quad \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau') \quad \Delta'; \Gamma' \vdash e_3 : \tau'}{\texttt{ifz}(e_1, \mathcal{C}, e_3) : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')}\ Cifz_2$$

$$\frac{\Delta'; \Gamma' \vdash e_1 : \texttt{int} \quad \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau') \quad \Delta'; \Gamma' \vdash e_2 : \tau'}{\texttt{ifz}(e_1, e_2, \mathcal{C}) : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')}\ Cifz_3$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma', f : \tau_1 \to \tau_2, x : \tau_1 \triangleright \tau_2) \quad \Delta' \vdash \tau_1\,\texttt{type}}{\texttt{fun}\ f(x : \tau_1).\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \to \tau_2)}\ Cfun$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta; f : \tau_1 \to \tau_2, x : \tau_1 \triangleright \tau_2) \quad \Delta' \vdash \tau_1\,\texttt{type}}{\widehat{\texttt{fun}}\ f(x : \tau_1).\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \Rightarrow \tau_2)}\ Cccfun$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \to \tau_2) \quad \Delta'; \Gamma' \vdash e : \tau_1}{\mathcal{C}\ e : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2)}\ Capp_1$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1) \quad \Delta'; \Gamma' \vdash e : \tau_1 \to \tau_2}{e\ \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2)}\ Capp_2$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \Rightarrow \tau_2) \quad \Delta'; \Gamma' \vdash e : \tau_1}{\mathcal{C}\,\hat{}\,e : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2)}\ Cccapp_1$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1) \quad \Delta'; \Gamma' \vdash e : \tau_1 \Rightarrow \tau_2}{e\,\hat{}\,\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2)}\ Cccapp_2$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1) \quad \Delta'; \Gamma' \vdash e : \tau_2}{\langle \mathcal{C}, e \rangle : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \times \tau_2)}\ Cpair_1$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2) \quad \Delta'; \Gamma' \vdash e : \tau_1}{\langle e, \mathcal{C} \rangle : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \times \tau_2)} \; \mathit{Cpair}_2$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_1 \times \tau_2) \quad i \in \{1, 2\}}{\pi_i \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_i)} \; \mathit{Cproj}$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta', \alpha; \Gamma' \triangleright \tau')}{\Lambda \alpha . \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \forall \alpha . \tau')} \; \mathit{Ctlam}$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \forall \alpha . \tau') \quad \Delta' \vdash \tau'' \, \texttt{type}}{\mathcal{C}[\tau''] : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright [\tau''/\alpha]\tau')} \; \mathit{Ctapp}$$

$$\frac{\Delta' \vdash \tau'' \, \texttt{type} \quad \Delta', \alpha \vdash \tau' \, \texttt{type} \quad \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright [\tau''/\alpha]\tau')}{\texttt{pack}[\tau'', \mathcal{C}] \, \texttt{as} \, \exists \alpha . \tau' : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \exists \alpha . \tau')} \; \mathit{Cpack}$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \exists \alpha . \tau_1) \quad \Delta', \alpha; \Gamma', x : \tau_1 \vdash e : \tau_2 \quad \Delta' \vdash \tau_2 \, \texttt{type}}{\texttt{unpack}[\alpha, x] = \mathcal{C} \, \texttt{in} \, e : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2)} \; \mathit{Cunpack}_1$$

$$\frac{\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta', \alpha; \Gamma', x : \tau_1 \triangleright \tau_2) \quad \Delta'; \Gamma' \vdash e : \exists \alpha . \tau_1 \quad \Delta' \vdash \tau_2 \, \texttt{type}}{\texttt{unpack}[\alpha, x] = e \, \texttt{in} \, \mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau_2)} \; \mathit{Cunpack}_2$$

4.1 Context Composition

Lemma 4.1. If $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$ and $\mathcal{C}' : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, then $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$.

Proof. By induction over the context typing rules on $\mathcal{C}'$.

Case for Cid
Assume that $\mathcal{C}' = o : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$. We know that $\mathcal{C}\{o\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$, which is equivalent to $o\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$, and therefore $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$, as required.

Case for $Cintop_1$
Assume that $\mathcal{C}' = \mathcal{C}''\,\mathtt{p}\,e : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \mathtt{int})$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \mathtt{int})$. Then by rule $Cintop_1$, we have that $\mathcal{C}''\{\mathcal{C}\{o\}\}\,\mathtt{p}\,e : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \mathtt{int})$, which is equivalent to $(\mathcal{C}''\,\mathtt{p}\,e)\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \mathtt{int})$, and so $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \mathtt{int})$.

Case for $Cintop_2$
Essentially the same as $Cintop_1$.

Case for $Cifz_1$
Assume that $\mathcal{C}' = \mathtt{ifz}(\mathcal{C}'', e_2, e_3) : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \mathtt{int})$. Then by rule $Cifz_1$, we have that $\mathtt{ifz}(\mathcal{C}''\{\mathcal{C}\{o\}\}, e_2, e_3) : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, which is equivalent to $\mathtt{ifz}(\mathcal{C}'', e_2, e_3)\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, and so $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$.

Case for $Cifz_2$
Assume that $\mathcal{C}' = \mathtt{ifz}(e_1, \mathcal{C}'', e_3) : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta'; \Gamma'' \triangleright \tau'')$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$. Then by rule $Cifz_1$, we have that $\mathtt{ifz}(e_1, \mathcal{C}''\{\mathcal{C}\{o\}\}, e_3) : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, which is equivalent to $\mathtt{ifz}(e_1, \mathcal{C}'', e_3)\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, and so $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$.

Case for $Cifz_3$
Essentially the same as $Cifz_2$.

Case for $Cfun$
Assume that $\mathcal{C}' = \mathtt{fun}\ f(x : \tau_1'').\mathcal{C}'' : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau_1'' \to \tau_2'')$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'', f : \tau_1'' \to \tau_2'', x : \tau_1'' \triangleright \tau_2'')$. Then by rule $Cfun$, we have that $\mathtt{fun}\ f(x : \tau_1'').\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau_1'' \to \tau_2'')$, which is equivalent to $(\mathtt{fun}\ f(x : \tau_1'').\mathcal{C}'')\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau_1'' \to \tau_2'')$, and so $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau_1'' \to \tau_2'')$.

Case for $Cccfun$
Assume that $\mathcal{C}' = \widehat{\mathtt{fun}}\ f(x : \tau_1'').\mathcal{C}'' : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau_1'' \Rightarrow \tau_2'')$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta; f : \tau_1'' \to \tau_2'', x : \tau_1'' \triangleright \tau_2'')$. Then by rule $Ccclam$, we have that $\widehat{\mathtt{fun}}\ f(x : \tau_1'').\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau_1'' \Rightarrow \tau_2'')$, which is equivalent to $(\widehat{\mathtt{fun}}\ f(x : \tau_1'').\mathcal{C}'')\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau_1'' \Rightarrow \tau_2'')$, and so $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau_1'' \Rightarrow \tau_2'')$.

Case for $Capp_1$
Assume that $\mathcal{C}' = \mathcal{C}''\ e : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_2'')$ and $\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_1'' \rightarrow \tau_2'')$. Since we know that $\Delta'';\Gamma'' \vdash e : \tau_1''$, by rule $Capp_1$, we have that $\mathcal{C}''\{\mathcal{C}\{o\}\}\ e : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_2'')$, which is equivalent to $(\mathcal{C}''\ e)\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_2'')$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_2'')$.

Case for $Capp_2$
Assume that $\mathcal{C}' = e\ \mathcal{C}'' : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_2'')$ and $\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma'' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_1'')$. Since we know that $\Delta'';\Gamma'' \vdash e : \tau_1'' \rightarrow \tau_2''$, by rule $Capp_2$, we have that $e\ \mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_2'')$, which is equivalent to $(e\ \mathcal{C}'')\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_2'')$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_2'')$

Case for $Cccapp_1$
Essentially the same as $Capp_1$.

Case for $Cccapp_2$
Essentially the same as $Capp_2$.

Case for $Cpair_1$
Assume that $\mathcal{C}' = \langle \mathcal{C}'', e_2 \rangle : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_1 \times \tau_2)$ and $\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_1)$. Then by $Cpair_1$ we have that $\langle \mathcal{C}''\{\mathcal{C}\{o\}\}, e_1 \rangle : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_1 \times \tau_2)$, which is equivalent to $\langle \mathcal{C}'', e_2 \rangle\{\mathcal{C}\{o\}\} : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_1 \times \tau_2)$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_1 \times \tau_2)$.

Case for $Cpair_2$
Similar to the case for $Cpair_1$.

Case for $Cproj$
Assume that $\mathcal{C}' = \pi_i \mathcal{C}'' : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_i)$ and $\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_1 \times \tau_2)$. Then by $Cproj$ we have that $\pi_i(\mathcal{C}''\{\mathcal{C}\{o\}\}) : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_i)$, which is equivalent to $(\pi_i \mathcal{C}'')\{\mathcal{C}\{o\}\} : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_i)$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau_i)$.

Case for $Ctlam$
Assume that $\mathcal{C}' = \Lambda\alpha.\mathcal{C}'' : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \forall\alpha.\tau'')$ and $\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'',\alpha;\Gamma'' \triangleright \tau'')$. Then by $Ctlam$ we have that $\Lambda\alpha.(\mathcal{C}''\{\mathcal{C}\{o\}\}) : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \forall\alpha.\tau'')$, which is equivalent to $(\Lambda\alpha.\mathcal{C}'')\{\mathcal{C}\{o\}\} : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \forall\alpha.\tau'')$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \forall\alpha.\tau'')$.

Case for $Ctapp$
Assume that $\mathcal{C}' = \mathcal{C}''[\tau_1] : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau'')$ and $\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright \forall\alpha.\tau'')$. Then by $Ctapp$ we have that $(\mathcal{C}''\{\mathcal{C}\{o\}\})[\tau_1] : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau'')$, which is equivalent to $(\mathcal{C}''[\tau_1])\{\mathcal{C}\{o\}\} : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau'')$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \tau'')$.

Case for $Cpack$
Assume that $\mathcal{C}' = \mathtt{pack}[\tau_1, \mathcal{C}'']\,\mathtt{as}\,\exists\alpha.\tau'' : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \exists\alpha.\tau'')$ and $\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta';\Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\Delta'';\Gamma'' \triangleright [\tau_1/\alpha]\tau'')$. Then by $Cpack$ we have that $\mathtt{pack}[\tau_1, \mathcal{C}''\{\mathcal{C}\{o\}\}]\,\mathtt{as}\,\exists\alpha.\tau'' : (\Delta';\Gamma' \triangleright \tau') \rightsquigarrow (\Delta'';\Gamma'' \triangleright \exists\alpha.\tau'')$, which is

equivalent to $(\texttt{pack}[\tau_1, \mathcal{C}''] \,\texttt{as}\, \exists\alpha.\tau'')\{\mathcal{C}\{o\}\} : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \exists\alpha.\tau'')$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \exists\alpha.\tau'')$.

Case for $Cunpack_1$
Assume that $\mathcal{C}' = \texttt{unpack}[\alpha, x] = \mathcal{C}'' \,\texttt{in}\, e : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta''; \Gamma'' \triangleright \exists\alpha.\tau_1)$. Then by $Cunpack_1$ we have that $\texttt{unpack}[\alpha, x] = (\mathcal{C}''\{\mathcal{C}\{o\}\}) \,\texttt{in}\, e : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, which is equivalent to $(\texttt{unpack}[\alpha, x] = \mathcal{C}'' \,\texttt{in}\, e)\{\mathcal{C}\{o\}\} : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$.

Case for $Cunpack_2$
Assume that $\mathcal{C}' = \texttt{unpack}[\alpha, x] = e \,\texttt{in}\, \mathcal{C}'' : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$. By induction on $\mathcal{C}''$ we get that $\mathcal{C}''\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'', \alpha; \Gamma'', x : \tau_1 \triangleright \tau_2)$. Then by $Cunpack_2$ we have that $\texttt{unpack}[\alpha, x] = e \,\texttt{in}(\mathcal{C}''\{\mathcal{C}\{o\}\}) : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, which is equivalent to $(\texttt{unpack}[\alpha, x] = e \,\texttt{in}\, \mathcal{C}'')\{\mathcal{C}\{o\}\} : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$, and thus $\mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\Delta''; \Gamma'' \triangleright \tau'')$.

□

5 Contextual Equivalence

5.1 Definitions

Definition A complete program e is the Kleene approximation of a complete program e', written $e \lesssim e'$, iff $e \downarrow \Rightarrow e' \downarrow$.

Definition Two complete programs, e and e', are Kleene equal, $e \simeq e'$, iff $e \downarrow \Leftrightarrow e' \downarrow$. Equivalently, if $e \lesssim e'$ and $e' \lesssim e$.

Clearly Kleene equivalence is reflexive, symmetric, and transitive and is thus an equivalence relation.

Definition Suppose that $\Delta; \Gamma \vdash e : \tau$ and $\Delta; \Gamma \vdash e' : \tau$ are two expressions of the same type. Then e contextually approximates e', written $\Delta; \Gamma \vdash e \leq e' : \tau$, iff $\mathcal{C}\{e\} \lesssim \mathcal{C}\{e'\}$ for every program context $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$. As a shorthand we just write $e \leq_\tau e'$ if $\cdot \vdash e : \tau$ and $\cdot \vdash e' : \tau$.

Clearly this is transitive and reflexive.

Definition Suppose that $\Delta; \Gamma \vdash e : \tau$ and $\Delta; \Gamma \vdash e' : \tau$ are two expressions of the same type. Two such expressions are *contextually equivalent*, written $\Delta; \Gamma \vdash e \cong e' : \tau$, iff $\mathcal{C}\{e\} \simeq \mathcal{C}\{e'\}$ for every program context $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$. Equivalently, if $\Delta; \Gamma \vdash e \leq e' : \tau$ and $\Delta; \Gamma \vdash e' \leq e : \tau$. As a shorthand we just write $e \cong_\tau e'$ if $\cdot \vdash e : \tau$ and $\cdot \vdash e' : \tau$.

Definition A family of equivalence relations is a *congruence* iff it is preserved by all contexts. Or equivalently, if $\Delta; \Gamma \vdash e \,\mathcal{E}\, e' : \tau$, then $\Delta'; \Gamma' \vdash \mathcal{C}\{e\} \,\mathcal{E}\, \mathcal{C}\{e'\} : \tau'$ for all contexts $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$.

Definition A family of equivalence relations is *consistent* iff $\cdot \vdash e \,\mathcal{E}\, e : \texttt{int}$ implies $e \simeq e'$.

Clearly contextual equivalence is both a congruence and consistent.

Definition A closing substitution γ for the context $\Gamma = x_1 : \tau_1, ..., x_n : \tau_n$ is a finite function that assigns the closed expressions $e_1 : \tau_1, ..., e_n : \tau_n$ to $x_1, ..., x_n$, respectively. We write $\widehat{\gamma}(e)$ for the substitution $[e_1/x_1]...[e_n/x_n]e$, and write $\gamma : \Gamma$ to mean that if $x : \tau$ occurs in Γ, then there exists a closed expression $e : \tau$ such that $\gamma(x) = e$. We write $\gamma \cong_\Gamma \gamma'$, where $\gamma : \Gamma$ and $\gamma' : \Gamma$, to say that $\gamma(x) \cong_{\Gamma(x)} \gamma'(x)$.

Lemma 5.1. If $e \cong_\tau e'$, then $e \simeq e'$.

Proof. Define the context $\mathcal{C} = (\lambda x : \tau.0)\ o$, where 0 is the integer zero. Clearly $\mathcal{C} : (\cdot \triangleright \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$. Since we know by assumption that $e \cong_\tau e' : \tau$, by definition we get that $\mathcal{C}\{e\} \simeq \mathcal{C}\{e'\}$, which is equivalent to

$$(\lambda x : \tau.0)\ e \simeq (\lambda x : \tau.0)\ e'$$

Suppose that e does not terminate. Then clearly $(\lambda x : \tau.0)\ e$ does not terminate either, by rule $Eapp_1$. Thus by the above, $(\lambda x : \tau.0)\ e'$ does not terminate. Then it must be the case that e' does not terminate, as otherwise $(\lambda x : \tau.0)\ e' \downarrow$, and we know that it does not.

Suppose that $e \downarrow$. Then clearly also $(\lambda x : \tau.0)\ e \downarrow$. Again by the above, this means that $(\lambda x : \tau.0)\ e' \downarrow$. Finally, by rule $Eapp_1$, it is clear that $e \downarrow$, as otherwise $(\lambda x : \tau.0)\ e'$ would not terminate. Therefore we have that $e \simeq e'$, as desired. □

Corollary 5.2. If $e \cong_\tau e'$, then $\mathcal{C}\{e\} \simeq \mathcal{C}\{e'\}$ for all $\mathcal{C} : (\cdot \triangleright \tau) \rightsquigarrow (\cdot \triangleright \tau')$.

Lemma 5.3. If $\Delta, \alpha; \Gamma \vdash e \cong e' : \tau$ and τ' $\texttt{type}$, then $\Delta; [\tau'/\alpha]\Gamma \vdash [\tau'/\alpha]e \cong [\tau'/\alpha]e' : [\tau'/\alpha]\tau$.

Proof. Let $\mathcal{C} : (\Delta; [\tau'/\alpha]\Gamma \triangleright [\tau'/\alpha]\tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$ be a program context. We need to show that

$$\mathcal{C}\{[\tau'/\alpha]e\} \simeq \mathcal{C}\{[\tau'/\alpha]e'\}$$

Since $\mathcal{C}$ is closed, this is equivalent to

$$[\tau'/\alpha]\mathcal{C}\{e\} \simeq [\tau'/\alpha]\mathcal{C}\{e'\}$$

Now define that context $\mathcal{C}' = (\Lambda\alpha.\mathcal{C}\{o\})[\tau'] : (\Delta, \alpha; \Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$. By assumption we know that $\mathcal{C}'\{e\} \simeq \mathcal{C}'\{e'\}$. However, $\mathcal{C}'\{e\} \simeq [\tau'/\alpha]\mathcal{C}\{e\}$ and $\mathcal{C}'\{e'\} \simeq [\tau'/\alpha]\mathcal{C}\{e'\}$, so therefore $\mathcal{C}\{[\tau'/\alpha]e\} \simeq \mathcal{C}\{[\tau'/\alpha]e'\}$ as desired. □

Lemma 5.4. If $\cdot; \Gamma \vdash e \cong e' : \tau$ and $\gamma : \Gamma$, then $\widehat{\gamma}(e) \cong_\tau \widehat{\gamma}(e')$. Also, if $\gamma \cong_\Gamma \gamma'$, then $\widehat{\gamma}(e) \cong_\tau \widehat{\gamma}'(e)$ and $\widehat{\gamma}(e') \cong_\tau \widehat{\gamma}'(e')$, as well as $\widehat{\gamma}(e) \cong_\tau \widehat{\gamma}'(e')$

Proof. Let $\mathcal{C} : (\cdot \triangleright \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$. Since $\cdot \vdash \widehat{\gamma}(e) : \tau$ and $\cdot \vdash \widehat{\gamma}(e') : \tau$, we are to show that $\mathcal{C}\{\widehat{\gamma}(e)\} \simeq \mathcal{C}\{\widehat{\gamma}(e')\}$. Define $\mathcal{C}_1$ to be the context

$$(\lambda x_1 : \tau_1 .... \lambda x_n : \tau_n . o)\ (e_1)\ ...\ (e_n)$$

where $\Gamma = x_1 : \tau_1, ..., x_n : \tau_n$ and $\gamma(x_1) = e_1, ..., \gamma(x_n) = e_n$. Clearly $\mathcal{C}_1 : (\Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \tau)$. Define the context $\mathcal{D}$ to be $\mathcal{C}\{\mathcal{C}_1\{o\}\}$. By Lemma 4.1, $\mathcal{D} : (\Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$. Therefore, since $\Gamma \vdash e \cong e' : \tau$, we know that $\mathcal{D}\{e\} \simeq \mathcal{D}\{e'\}$. But by construction, $\mathcal{D}\{e\} \simeq \mathcal{C}\{\widehat{\gamma}(e)\}$ and $\mathcal{D}\{e'\} \simeq \mathcal{C}\{\widehat{\gamma}(e')\}$, so $\mathcal{C}\{\widehat{\gamma}(e)\} \simeq \mathcal{C}\{\widehat{\gamma}(e')\}$. Since $\mathcal{C}$ is arbitrary, we have that $\widehat{\gamma}(e) \cong \widehat{\gamma}(e')$.

Now, define $\mathcal{C}_1'$ to be the context

$$(\lambda x_1 : \tau_1 .... \lambda x_n : \tau_n . o)\ (e_1')\ ...\ (e_n')$$

where $\gamma'(x_1) = e_1', ..., \gamma'(x_n) = e_n'$. Defining $\mathcal{D}'$ to be $\mathcal{C}\{\mathcal{C}_1'\{o\}\}$, by the same reasoning as above we can get that $\mathcal{D}'\{e\} \simeq \mathcal{D}'\{e'\}$ as well as $\widehat{\gamma}'(e) \cong \widehat{\gamma}'(e')$. Assuming that $\gamma \cong_\Gamma \gamma'$, we have by congruence that $\mathcal{D}\{e\} \cong_{\texttt{int}} \mathcal{D}'\{e\}$ and $\mathcal{D}\{e'\} \cong_{\texttt{int}} \mathcal{D}'\{e'\}$. By the definition of contextual equivalence and applying the identity context, we have that $\mathcal{D}\{e\} \simeq \mathcal{D}'\{e\}$ and $\mathcal{D}\{e'\} \simeq \mathcal{D}'\{e'\}$. By the same logic as above, we can show that $\widehat{\gamma}(e) \cong_\tau \widehat{\gamma}'(e)$ and $\widehat{\gamma}(e') \cong_\tau \widehat{\gamma}'(e')$, as desired. Since we also know that $\mathcal{D}'\{e\} \simeq \mathcal{D}'\{e'\}$ as mentioned above, by transitivity of Kleene equality and using the fact that $\mathcal{D}\{e\} \simeq \mathcal{D}'\{e\}$, we get that $\mathcal{D}\{e\} \simeq \mathcal{D}'\{e'\}$. This implies that $\widehat{\gamma}(e) \cong_\tau \widehat{\gamma}'(e')$ by the same reasoning as above, as desired. □

Lemma 5.5. If $\Delta; \Gamma \vdash e \leq e' : \tau$ and $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$, then $\Delta'; \Gamma' \vdash \mathcal{C}\{e\} \leq \mathcal{C}\{e'\} : \tau'$.

Proof. Let $\mathcal{C}' : (\Delta'; \Gamma' \triangleright \tau') \rightsquigarrow (\cdot \triangleright \texttt{int})$ be an arbitrary context and $\mathcal{C}$ be defined as above. Define a new context $\mathcal{C}'' = \mathcal{C}'\{\mathcal{C}\{o\}\} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$ by Lemma 4.1. Then, since $\Delta; \Gamma \vdash e \leq e' : \tau$, we know that $\mathcal{C}''\{e\} \lesssim \mathcal{C}''\{e'\}$, or equivalently, $\mathcal{C}'\{\mathcal{C}\{e\}\} \lesssim \mathcal{C}'\{\mathcal{C}\{e'\}\}$. But since $\mathcal{C}'$ was arbitrary, we have that $\Delta'; \Gamma' \vdash \mathcal{C}\{e\} \leq \mathcal{C}\{e'\} : \tau'$, as desired. □

Corollary 5.6. If $\Gamma \vdash e \cong e' : \tau$ and $\mathcal{C} : (\Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$, then $\Delta'; \Gamma' \vdash \mathcal{C}\{e\} \cong \mathcal{C}\{e'\} : \tau'$.

Lemma 5.7. (Unproven Assumption) If $\Delta; \Gamma \vdash e : \tau$ and $e \mapsto e'$, then $\Delta; \Gamma \vdash e \cong e' : \tau$.

5.2 Substitutivity

Lemma 5.8. If $\Delta; \Gamma \vdash e_1 \leq e_2 : \tau'$ and $\Delta; \Gamma, x : \tau' \vdash e : \tau$, then $\Delta; \Gamma \vdash [e_1/x]e \leq [e_2/x]e : \tau$.

Proof. By induction on the structure of e.

Case for $e = \texttt{()}$
If $e = \texttt{()}$, then we just need to show $\Delta; \Gamma \vdash [e_1/x]\texttt{()} \leq [e_2/x]\texttt{()} : \texttt{unit}$, which is equivalent to $\Delta; \Gamma \vdash \texttt{()} \leq \texttt{()} : \texttt{unit}$, which is trivially true.

Case for $e = x$
If $e = x$, then we just need to show $\Delta; \Gamma \vdash [e_1/x]x \leq [e_2/x]x : \tau$, which is equivalent to $\Delta; \Gamma \vdash e_1 \leq e_2 : \tau$, which we already know by assumption (and since $\tau = \tau'$ in this case).

Case for $e = n$
If $e = n$, then we just need to show $\Delta; \Gamma \vdash [e_1/x]n \leq [e_2/x]n : \texttt{int}$, which is equivalent to $\Delta; \Gamma \vdash n \leq n : \texttt{int}$, which is trivially true.

Case for $e = e'_1 \, \texttt{p} \, e'_2$
If $e = e'_1 \, \texttt{p} \, e'_2$, then we just need to show $\Delta; \Gamma \vdash [e_1/x](e'_1 \, \texttt{p} \, e'_2) \leq [e_2/x](e'_1 \, \texttt{p} \, e'_2) : \texttt{int}$, which is equivalent to

$$\Delta; \Gamma \vdash ([e_1/x]e'_1) \, \texttt{p}([e_1/x]e'_2) \leq ([e_2/x]e'_1) \, \texttt{p}([e_2/x]e'_2) : \texttt{int}$$

By induction, we get that $\Delta; \Gamma \vdash [e_1/x]e'_1 \leq [e_2/x]e'_1 : \texttt{int}$ and $\Delta; \Gamma \vdash [e_1/x]e'_2 \leq [e_2/x]e'_2 : \texttt{int}$. If we define the context $\mathcal{C}_1 = o \, \texttt{p}([e_1/x]e'_2)$, then we can apply Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e'_1 \leq [e_2/x]e'_1 : \texttt{int}$ and get that

$$\Delta; \Gamma \vdash ([e_1/x]e'_1) \, \texttt{p}([e_1/x]e'_2) \leq ([e_2/x]e'_1) \, \texttt{p}([e_1/x]e'_2) : \texttt{int}$$

Similarly, define $\mathcal{C}_2 = ([e_2/x]e'_1) \, \texttt{p} \, o$ and apply Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e'_2 \leq [e_2/x]e'_2 : \texttt{int}$ to get that

$$\Delta; \Gamma \vdash ([e_2/x]e'_1) \, \texttt{p}([e_1/x]e'_2) \leq ([e_2/x]e'_1) \, \texttt{p}([e_2/x]e'_2) : \texttt{int}$$

Finally, by transitivity, we get the desired result

$$\Delta; \Gamma \vdash ([e_1/x]e'_1) \, \texttt{p}([e_1/x]e'_2) \leq ([e_2/x]e'_1) \, \texttt{p}([e_2/x]e'_2) : \texttt{int}$$

Case for $e = \texttt{ifz}(e'_1, e'_2, e'_3)$
If $e = \texttt{ifz}(e'_1, e'_2, e'_3)$, then we just need to show $\Delta; \Gamma \vdash [e_1/x]\texttt{ifz}(e'_1, e'_2, e'_3) \leq [e_2/x]\texttt{ifz}(e'_1, e'_2, e'_3) : \tau$, which is equivalent to

$$\Delta; \Gamma \vdash \texttt{ifz}([e_1/x]e'_1, [e_1/x]e'_2, [e_1/x]e'_3) \leq \texttt{ifz}([e_2/x]e'_1, [e_2/x]e'_2, [e_2/x]e'_3) : \tau$$

By induction on the subterms, we get that $\Delta; \Gamma \vdash [e_1/x]e'_1 \leq [e_2/x]e'_1 : \texttt{int}$, $\Delta; \Gamma \vdash [e_1/x]e'_2 \leq [e_2/x]e'_2 : \tau$, and $\Delta; \Gamma \vdash [e_1/x]e'_3 \leq [e_2/x]e'_3 : \tau$. Define the context $\mathcal{C}_1 = \texttt{ifz}(o, [e_1/x]e'_2, [e_1/x]e'_3)$. Applying Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e'_1 \leq [e_2/x]e'_1 : \texttt{int}$, we get that

$$\Delta; \Gamma \vdash \texttt{ifz}([e_1/x]e'_1, [e_1/x]e'_2, [e_1/x]e'_3) \leq \texttt{ifz}([e_2/x]e'_1, [e_1/x]e'_2, [e_1/x]e'_3) : \tau$$

Again using Corollary 5.6 but on $\Delta; \Gamma \vdash [e_1/x]e'_2 \leq [e_2/x]e'_2 : \texttt{int}$ and the context $\mathcal{C}_2 = \texttt{ifz}([e_2/x]e'_1, o, [e_1/x]e'_3)$, we get that

$$\Delta; \Gamma \vdash \texttt{ifz}([e_2/x]e'_1, [e_1/x]e'_2, [e_1/x]e'_3) \leq \texttt{ifz}([e_2/x]e'_1, [e_2/x]e'_2, [e_1/x]e'_3) : \tau$$

Finally, applying Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e'_3 \leq [e_2/x]e'_3 : \texttt{int}$ and the context $\mathcal{C}_3 = \texttt{ifz}([e_2/x]e'_1, [e_2/x]e'_2, o)$, we get

$$\Delta; \Gamma \vdash \texttt{ifz}([e_2/x]e'_1, [e_2/x]e'_2, [e_1/x]e'_3) \leq \texttt{ifz}([e_2/x]e'_1, [e_2/x]e'_2, [e_2/x]e'_3) : \tau$$

Then using the transitivity of contextual equivalence, we get that

$$\Delta; \Gamma \vdash \texttt{ifz}([e_1/x]e'_1, [e_1/x]e'_2, [e_1/x]e'_3) \leq \texttt{ifz}([e_2/x]e'_1, [e_2/x]e'_2, [e_2/x]e'_3) : \tau$$

Case for $e = \texttt{fun}\ f(y : \tau').e'$
If $e = \texttt{fun}\ f(y : \tau').e'$, then we just need to show

$$\Delta; \Gamma \vdash [e_1/x](\texttt{fun}\ f(y : \tau').e') \leq [e_2/x](\texttt{fun}\ f(y : \tau').e') : \tau' \to \tau$$

which is equivalent to $\Delta; \Gamma \vdash \texttt{fun}\ f(y : \tau').[e_1/x]e' \leq \texttt{fun}\ f(y : \tau').[e_2/x]e' : \tau' \to \tau$. By induction we know that $\Delta; \Gamma, f : \tau' \to \tau, y : \tau' \vdash [e_1/x]e' \leq [e_1/x]e' : \tau$, so we can apply Corollary 5.6 to this with the context $\mathcal{C} = \texttt{fun}\ f(y : \tau').o : (\Delta; \Gamma, f : \tau' \to \tau, y : \tau' \triangleright \tau) \rightsquigarrow (\Delta; \Gamma \triangleright \tau' \to \tau)$ to get that

$$\Delta; \Gamma \vdash \texttt{fun}\ f(y : \tau').[e_1/x]e' \leq \texttt{fun}\ f(y : \tau').[e_2/x]e' : \tau' \to \tau$$

Case for $e = e'_1\ e'_2$
If $e = e'_1\ e'_2$, then we just need to show

$$\Delta; \Gamma \vdash [e_1/x](e'_1\ e'_2) \leq [e_2/x](e'_1\ e'_2) : \tau$$

which is equivalent to

$$\Delta; \Gamma \vdash ([e_1/x]e'_1)\ ([e_1/x]e'_2) \leq ([e_2/x]e'_1)\ ([e_2/x]e'_2) : \tau$$

By induction we get that $\Delta; \Gamma \vdash [e_1/x]e'_1 \leq [e_2/x]e'_1 : \tau' \to \tau$ and $\Delta; \Gamma \vdash [e_1/x]e'_2 \leq [e_2/x]e'_2 : \tau'$. Applying Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e'_1 \leq [e_2/x]e'_1 : \tau' \to \tau$ the context $\mathcal{C}_1 = o\ ([e_1/x]e'_2)$, we get that

$$\Delta; \Gamma \vdash ([e_1/x]e'_1)\ ([e_1/x]e'_2) \leq ([e_2/x]e'_1)\ ([e_1/x]e'_2) : \tau$$

Similarly, applying Corollary 5.6 to $[e_1/x]e'_2 \leq [e_2/x]e'_2 : \tau'$ and the context $\mathcal{C}_2 = ([e_2/x]e'_1)\ o$, we get that

$$\Delta; \Gamma \vdash ([e_2/x]e'_1)\ ([e_1/x]e'_2) \leq \tau([e_2/x]e'_1)\ ([e_2/x]e'_2) : \tau$$

Thus by transitivity, we have the desired result:

$$\Delta; \Gamma \vdash ([e_1/x]e'_1)\ ([e_1/x]e'_2) \leq ([e_2/x]e'_1)\ ([e_2/x]e'_2) : \tau$$

Case for $e = \widehat{\texttt{fun}}\ f(y : \tau).e'$
If $e = \widehat{\texttt{fun}}\ f(y : \tau').e'$, then we just need to show

$$\Delta; \Gamma \vdash [e_1/x](\widehat{\texttt{fun}}\ f(y : \tau').e') \leq [e_2/x](\widehat{\texttt{fun}}\ f(y : \tau').e') : \tau' \Rightarrow \tau$$

However, since this is a closed function, we know that this is equivalent to

$$\Delta; \Gamma \vdash \widehat{\texttt{fun}}\ f(y : \tau').e' \leq \widehat{\texttt{fun}}\ f(y : \tau').e' : \tau' \Rightarrow \tau$$

which we already know by reflexivity.

Case for $e = e_1'\frown e_2'$
If $e = e_1'\frown e_2'$, then we just need to show

$$\Delta; \Gamma \vdash [e_1/x](e_1'\frown e_2') \leq [e_2/x](e_1'\frown e_2') : \tau$$

which is equivalent to

$$\Delta; \Gamma \vdash ([e_1/x]e_1')\frown([e_1/x]e_2') \leq ([e_2/x]e_1')\frown([e_2/x]e_2') : \tau$$

By induction we get that $\Delta; \Gamma \vdash [e_1/x]e_1' \leq [e_2/x]e_1' : \tau' \Rightarrow \tau$ and $\Delta; \Gamma \vdash [e_1/x]e_2' \leq [e_2/x]e_2' : \tau'$. Applying Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e_1' \leq [e_2/x]e_1' : \tau' \Rightarrow \tau$ the context $\mathcal{C}_1 = o\frown([e_1/x]e_2')$, we get that

$$\Delta; \Gamma \vdash ([e_1/x]e_1')\frown([e_1/x]e_2') \leq ([e_2/x]e_1')\frown([e_1/x]e_2') : \tau$$

Similarly, applying Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e_2' \leq [e_2/x]e_2' : \tau'$ and the context $\mathcal{C}_2 = ([e_2/x]e_1')\frown o$, we get that

$$\Delta; \Gamma \vdash ([e_2/x]e_1')\frown([e_1/x]e_2') \leq ([e_2/x]e_1')\frown([e_2/x]e_2') : \tau$$

Thus by transitivity, we have the desired result:

$$\Delta; \Gamma \vdash ([e_1/x]e_1')\frown([e_1/x]e_2') \leq ([e_2/x]e_1')\frown([e_2/x]e_2') : \tau$$

Case for $e = \langle e_1', e_2' \rangle$
If $e = \langle e_1', e_2' \rangle$, then we just need to show

$$\Delta; \Gamma \vdash [e_1/x]\langle e_1', e_2' \rangle \leq [e_2/x]\langle e_1', e_2' \rangle : \tau_1 \times \tau_2$$

which is equivalent to

$$\Delta; \Gamma \vdash \langle [e_1/x]e_1', [e_1/x]e_2' \rangle \leq \langle [e_2/x]e_1', [e_2/x]e_2' \rangle : \tau_1 \times \tau_2$$

By induction we get that $\Delta; \Gamma \vdash [e_1/x]e_1' \leq [e_2/x]e_1' : \tau_1$ and $\Delta; \Gamma \vdash [e_1/x]e_2' \leq [e_2/x]e_2' : \tau_2$. Applying Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e_1' \leq [e_2/x]e_1' : \tau_1$ the context $\mathcal{C}_1 = \langle o, [e_1/x]e_2' \rangle$, we get that

$$\Delta; \Gamma \vdash \langle [e_1/x]e_1', [e_1/x]e_2' \rangle \leq \langle [e_2/x]e_1', [e_1/x]e_2' \rangle : \tau_1 \times \tau_2$$

Similarly, applying Corollary 5.6 to $\Delta; \Gamma \vdash [e_1/x]e_2' \leq [e_2/x]e_2' : \tau_2$ and the context $\mathcal{C}_2 = \langle [e_2/x]e_1', o \rangle$, we get that

$$\Delta; \Gamma \vdash \langle [e_2/x]e_1', [e_1/x]e_2' \rangle \leq \langle [e_2/x]e_1', [e_2/x]e_2' \rangle : \tau_1 \times \tau_2$$

Thus by transitivity, we have the desired result:

$$\Delta; \Gamma \vdash \langle [e_1/x]e_1', [e_1/x]e_2' \rangle \leq \langle [e_2/x]e_1', [e_2/x]e_2' \rangle : \tau_1 \times \tau_2$$

Case for $e = \pi_i e'$
If $e = \pi_i e'$, then we just need to show

$$\Delta; \Gamma \vdash [e_1/x]\pi_i e' \leq [e_2/x]\pi_i e' : \tau_i$$

which is equivalent to

$$\Delta; \Gamma \vdash \pi_i([e_1/x]e') \leq \pi_i([e_2/x]e') : \tau_i$$

By induction we get that $[e_1/x]e_1' \leq [e_2/x]e_1' : \tau_1 \times \tau_2$. Applying Corollary 5.6 to this with the context $\mathcal{C} = \pi_i o$, we get the desired result

$$\Delta; \Gamma \vdash \pi_i([e_1/x]e') \leq \pi_i([e_2/x]e') : \tau_i$$

Case for $e = \Lambda\alpha.e'$
If $e = \Lambda\alpha.e'$, then we just need to show

$$\Delta;\Gamma \vdash [e_1/x]\Lambda\alpha.e' \leq [e_2/x]\Lambda\alpha.e' : \forall\alpha.\tau'$$

which is equivalent to

$$\Delta;\Gamma \vdash \Lambda\alpha.[e_1/x]e' \leq \Lambda\alpha.[e_2/x]e' : \forall\alpha.\tau'$$

since neither e_1 nor e_2 depend on α. By induction we get that $\Delta,\alpha;\Gamma \vdash [e_1/x]e' \leq [e_2/x]e' : \tau'$. Applying Corollary 5.6 to this with the context $\mathcal{C} = \Lambda\alpha.o$, we get the desired result

$$\Delta;\Gamma \vdash [e_1/x]\Lambda\alpha.e' \leq [e_2/x]\Lambda\alpha.e' : \forall\alpha.\tau'$$

Case for $e = e'[\tau']$
If $e = e'[\tau']$, then we just need to show

$$\Delta;\Gamma \vdash [e_1/x](e'[\tau']) \leq [e_2/x](e'[\tau']) : [\tau'/\alpha]\tau$$

which is equivalent to

$$\Delta;\Gamma \vdash ([e_1/x]e')[\tau'] \leq ([e_2/x]e')[\tau'] : [\tau'/\alpha]\tau$$

By induction we get that $\Delta;\Gamma \vdash [e_1/x]e' \leq [e_2/x]e' : \forall\alpha.\tau$. Applying Corollary 5.6 to this with the context $\mathcal{C} = o[\tau']$, we get the desired result

$$\Delta;\Gamma \vdash ([e_1/x]e')[\tau'] \leq ([e_2/x]e')[\tau'] : [\tau'/\alpha]\tau$$

Case for $e = \texttt{pack}[\tau', e']\,\texttt{as}\,\exists\alpha.\tau$
If $e = \texttt{pack}[\tau', e']\,\texttt{as}\,\exists\alpha.\tau$, then we just need to show that

$$\Delta;\Gamma \vdash [e_1/x](\texttt{pack}[\tau', e']\,\texttt{as}\,\exists\alpha.\tau) \leq [e_2/x](\texttt{pack}[\tau', e']\,\texttt{as}\,\exists\alpha.\tau) : \exists\alpha\tau$$

which is equivalent to

$$\Delta;\Gamma \vdash \texttt{pack}[\tau', [e_1/x]e']\,\texttt{as}\,\exists\alpha.\tau \leq \texttt{pack}[\tau', [e_2/x]e']\,\texttt{as}\,\exists\alpha.\tau : \exists\alpha\tau$$

By induction we get that $\Delta;\Gamma \vdash [e_1/x]e' \leq [e_2/x]e' : [\tau'/\alpha]\tau$. Applying Corollary 5.6 to this with the context $\mathcal{C} = \texttt{pack}[\tau', o]\,\texttt{as}\,\exists\alpha.\tau$, we get the desired result

$$\Delta;\Gamma \vdash \texttt{pack}[\tau', [e_1/x]e']\,\texttt{as}\,\exists\alpha.\tau \leq \texttt{pack}[\tau', [e_2/x]e']\,\texttt{as}\,\exists\alpha.\tau : \exists\alpha\tau$$

Case for $e = \texttt{unpack}[\alpha, y] = e'_1\,\texttt{in}\,e'_2$
If $e = \texttt{unpack}[\alpha, y] = e'_1\,\texttt{in}\,e'_2$, then we need to show that

$$\Delta;\Gamma \vdash [e_1/x](\texttt{unpack}[\alpha, y] = e'_1\,\texttt{in}\,e'_2) \leq [e_2/x](\texttt{unpack}[\alpha, y] = e'_1\,\texttt{in}\,e'_2) : \tau_2$$

which is equivalent to

$$\Delta;\Gamma \vdash \texttt{unpack}[\alpha, y] = [e_1/x]e'_1\,\texttt{in}[e_1/x]e'_2 \leq \texttt{unpack}[\alpha, y] = [e_2/x]e'_1\,\texttt{in}[e_2/x]e'_2 : \tau_2$$

By induction we get that $\Delta;\Gamma \vdash [e_1/x]e'_1 \leq [e_2/x]e'_1 : \exists\alpha.\tau_1$ and $\Delta,\alpha;\Gamma, x : \tau_1 \vdash [e_1/x]e'_2 \leq [e_2/x]e'_2 : \tau_2$. Applying Corollary 5.6 to the first of these with the context $\mathcal{C} = (\texttt{unpack}[\alpha, y] = o\,\texttt{in}[e_1/x]e'_2)$, we get that

$$\Delta;\Gamma \vdash \texttt{unpack}[\alpha, y] = [e_1/x]e'_1\,\texttt{in}[e_1/x]e'_2 \leq \texttt{unpack}[\alpha, y] = [e_2/x]e'_1\,\texttt{in}[e_1/x]e'_2 : \tau_2$$

Applying Corollary 5.6 to the second inductive result with the context
$\mathcal{C} = (\texttt{unpack}[\alpha, y] = [e_2/x]e'_1\,\texttt{in}\,o)$, we get that

$$\Delta;\Gamma \vdash \texttt{unpack}[\alpha, y] = [e_2/x]e'_1\,\texttt{in}[e_1/x]e'_2 \leq \texttt{unpack}[\alpha, y] = [e_2/x]e'_1\,\texttt{in}[e_2/x]e'_2 : \tau_2$$

The desired result then follows from the above two results by transitivity.

□

Corollary 5.9. If $\Delta; \Gamma \vdash e_1 \cong e_2 : \tau'$ and $\Delta; \Gamma, x : \tau' \vdash e : \tau$, then $\Delta; \Gamma \vdash [e_1/x]e \cong [e_2/x]e : \tau$.

6 Compactness

Definition Define the "unwindings" of a recursive function $f = \texttt{fun}\ g(x : \tau).e$ as

$$\texttt{fun}^0\ g(x : \tau).e = \texttt{fun}\ g(x : \tau).\bot$$

$$\texttt{fun}^{i+1}\ g(x : \tau).e = \texttt{fun}\ g(x : \tau).[\texttt{fun}^i\ g(x : \tau).e/g]e$$

As a shorthand, we write $f^i = \texttt{fun}^i\ g(x : \tau).e$ and $f^w = f$.

Similarly, define the "unwindings" of a recursive closure-converted function $f = \widehat{\texttt{fun}}\ g(x : \tau).e$ as

$$\widehat{\texttt{fun}}^0\ g(x : \tau).e = \widehat{\texttt{fun}}\ g(x : \tau).\bot$$

$$\widehat{\texttt{fun}}^{i+1}\ g(x : \tau).e = \widehat{\texttt{fun}}\ g(x : \tau).[\widehat{\texttt{fun}}^i\ g(x : \tau).e/g]e$$

As a shorthand, we write $f^i = \widehat{\texttt{fun}}^i\ g(x : \tau).e$ and $f^w = f$.

Clearly in either case f^0 diverges by definition, and f^i behaves as a function that can only be called i times before looping forever. Note that when using f, unless specified, it could be referring to either type of function.

Also, as another shorthand, we write $e^{f[o]} = [f^o/w]e$ for some $\cdot \vdash f : \tau \to \tau'$ and $w : \tau \to \tau' \vdash e$, where o can be filled by w or any i.

Lemma 6.1. (Unproven Assumption) For all $\Gamma \vdash e : \tau$, $\Gamma \vdash \bot \leq e : \tau$ for some $\bot$ such that $\Gamma \vdash \bot : \tau$.

Lemma 6.2. For a given function $f = \texttt{fun}\ g(x : \tau').e$ or $f = \widehat{\texttt{fun}}\ g(x : \tau').e$ and all i, j such that $0 \leq i \leq j$, $f^i \leq_\tau f^j$.

Proof. We proceed by induction to first show that for all $0 \leq i$, $f^i \leq_\tau f^{i+1}$.

Case for $i = 0$
Suppose that $f = \texttt{fun}\ g(x : \tau').e$, with $\cdot \vdash f : \tau' \to \tau$. By their definitions, we know that $f^0 = \texttt{fun}\ g(x : \tau').\bot$ and $f^1 = \texttt{fun}\ g(x : \tau').[\texttt{fun}\ g(x : \tau').\bot/g]e$. By Lemma 6.1 we know that

$$g : \tau' \to \tau, x : \tau' \vdash \bot \leq [\texttt{fun}\ g(x : \tau').\bot/g]e : \tau$$

Then using the context $\mathcal{C} = \texttt{fun}\ g(x : \tau').o$, where $\mathcal{C} : (g : \tau' \to \tau, x : \tau' \triangleright \tau) \rightsquigarrow (\cdot \triangleright \tau' \to \tau)$, we can apply Lemma 5.5 to get that

$$\mathcal{C}\{\bot\} \leq_{\tau' \to \tau} \mathcal{C}\{[\texttt{fun}\ g(x : \tau').\bot/g]e\}$$

which is equivalent to

$$\texttt{fun}\ g(x : \tau').\bot \leq_{\tau' \to \tau} \texttt{fun}\ g(x : \tau').[\texttt{fun}\ g(x : \tau').\bot/g]e$$

which is the same as $f^0 \leq_{\tau' \to \tau} f^1$ and is what we wanted to show.

The other possibility is that $f = \widehat{\texttt{fun}}\ g(x : \tau').e$, with $\cdot \vdash f : \tau' \Rightarrow \tau$. By their definitions, we know that $f^0 = \widehat{\texttt{fun}}\ g(x : \tau').\bot$ and $f^1 = \widehat{\texttt{fun}}\ g(x : \tau').[\widehat{\texttt{fun}}\ g(x : \tau').\bot/g]e$. By Lemma 6.1 we know that

$$g : \tau' \Rightarrow \tau, x : \tau' \vdash \bot \leq [\widehat{\texttt{fun}}\ g(x : \tau').\bot/g]e : \tau$$

Then using the context $\mathcal{C} = \widehat{\texttt{fun}}\ g(x : \tau').o$, where $\mathcal{C} : (g : \tau' \Rightarrow \tau, x : \tau' \triangleright \tau) \rightsquigarrow (\cdot \triangleright \tau' \Rightarrow \tau)$, we can apply Lemma 5.5 to get that

$$\mathcal{C}\{\bot\} \leq_{\tau' \Rightarrow \tau} \mathcal{C}\{[\widehat{\texttt{fun}}\ g(x : \tau').\bot/g]e\}$$

which is equivalent to

$$\widehat{\texttt{fun}}\ g(x : \tau').\bot \leq_{\tau' \Rightarrow \tau} \widehat{\texttt{fun}}\ g(x : \tau').[\widehat{\texttt{fun}}\ g(x : \tau').\bot/g]e$$

which is the same as $f^0 \leq_{\tau' \Rightarrow \tau} f^1$ and is what we wanted to show.

Case for $i > 0$
We want to show that $f^i \leq_\tau f^{i+1}$. There are two cases to consider.

If $f = \texttt{fun}\ g(x : \tau).e$, then clearly by definition $f^i = \lambda x : \tau'.[f^{i-1}/f]e$ and similarly $f^{i+1} = \lambda x : \tau'.[f^i/f]e$. By induction we know that $f^{i-1} \leq_\tau f^i$, so by Lemma 5.8 we get that $[f^{i-1}/f]e \leq_\tau [f^i/f]e$. Then using the context $\mathcal{C} = \lambda x : \tau'.o$, we can apply Lemma 5.5 to get that

$$\lambda x : \tau'.[f^{i-1}/f]e \leq_\tau \lambda x : \tau'.[f^i/f]e$$

and thus we have the desired result, that $f^i \leq_\tau f^{i+1}$.

If $f = \widehat{\texttt{fun}}\ g(x : \tau).e$, then clearly by definition $f^i = \widehat{\lambda} x : \tau'.[f^{i-1}/f]e$ and similarly $f^{i+1} = \widehat{\lambda} x : \tau'.[f^i/f]e$. By induction we know that $f^{i-1} \leq_\tau f^i$, so by Lemma 5.8 we get that $[f^{i-1}/f]e \leq_\tau [f^i/f]e$. Then using the context $\mathcal{C} = \widehat{\lambda} x : \tau'.o$, we can apply Lemma 5.5 to get that

$$\widehat{\lambda} x : \tau'.[f^{i-1}/f]e \leq_\tau \widehat{\lambda} x : \tau'.[f^i/f]e$$

and thus we have the desired result, that $f^i \leq_\tau f^{i+1}$.

Now we want to show that for all i, j such that $0 \leq i \leq j$, $f^i \leq_\tau f^j$. We proceed by induction on j.

Case for $j = i$
Trivially true, as $f^i = f^j$, so clearly $f^i \leq_\tau f^j$.

Case for $j > i$
By induction we get that $f^i \leq_\tau f^{j-1}$. By our proof above we know that $f^{j-1} \leq_\tau f^j$, and so by transitivity we know that $f^i \leq_\tau f^j$.

□

Corollary 6.3. For a given function $f = \texttt{fun}\ g(x : \tau').e$ or $f = \widehat{\texttt{fun}}\ g(x : \tau').e$ and all $i \geq 0$, $f^i \leq_\tau f^w$.

Lemma 6.4.

(1) $f = \texttt{fun}\ g(x : \tau_1).e'$ and $w : \tau_1 \to \tau_2 \vdash e : \tau$

(2) $f = \widehat{\texttt{fun}}\ g(x : \tau_1).e'$ and $w : \tau_1 \Rightarrow \tau_2 \vdash e : \tau$

For all i, j such that $0 \leq i \leq j$, if either (1) or (2) hold, then $e^{f[i]} \leq_\tau e^{f[j]}$

Proof. Follows immediately from the above Lemma 6.3 and Lemma 5.8. □

6.1 Simulation

Lemma 6.5.

$$(1) \qquad f = \texttt{fun}\ g(x : \tau_1).e' \text{ where } \cdot \vdash f : \tau_1 \to \tau_2 \text{ and } \Gamma = w : \tau_1 \to \tau_2$$

$$(2) \qquad f = \widehat{\texttt{fun}}\ g(x : \tau_1).e' \text{ where } \cdot \vdash f : \tau_1 \Rightarrow \tau_2 \text{ and } \Gamma = w : \tau_1 \Rightarrow \tau_2$$

Given either (1) or (2) from above, and that $\Gamma \vdash e : \tau$ and $e^{f[w]} \mapsto^i v$ (where i indicates that it takes exactly i steps), then $\exists j, v'$ such that $\Gamma \vdash v' : \tau$, $v = v'^{f[w]}$, and $\forall k \geq j. e^{f[k]} \geq v'^{f[k-j]}$

Proof. By induction on the length of $e^{f[w]} \mapsto^i v$, with an inner induction on the structure of e.

Case for $i = 0$
In this case, it must be that $e^{f[w]}\ \texttt{val}$. There are a few cases for this:

Case for $e = w$
We must consider this case because e has w unbound in it, and we are substituting a value in for w. Thus we have that $e^{f[w]} = f^w = v$ and so $e^{f[k]} = f^k$. Let $j = 0$ and $v' = w$. There are two cases based on the two possibilities mentioned at the beginning, (1) and (2).

If (1) is the case, then we have that $v = f^w = v'^{f[w]}$, as desired. Also, for some k, $v'^{f[k-j]} = v'^{f[k]} = \texttt{fun}^k\ g(x : \tau_1).e'$, as well as $e^{f[k]} = \texttt{fun}^k\ g(x : \tau_1).e'$. Thus, since $v'^{f[k-j]} = e^{f[k]}$, clearly $v'^{f[k-j]} \leq_{\tau_1 \to \tau_2} e^{f[k]}$ for all k, since k was arbitrary. Thus the result holds in this case.

If (2) is the case, then we have that $v = f^w = v'^{f[w]}$, as desired. Also, letting k be arbitrary, $v'^{f[k-j]} = v'^{f[k]} = \widehat{\texttt{fun}}^k\ g(x : \tau_1).e'$, as well as $e^{f[k]} = \widehat{\texttt{fun}}^k\ g(x : \tau_1).e'$. Thus, since $v'^{f[k-j]} = e^{f[k]}$, clearly $v'^{f[k-j]} \leq_{\tau_1 \to \tau_2} e^{f[k]}$ for all k, since k was arbitrary. Thus the result holds in this case.

Case for $e = \texttt{()}$
In this situation, $e^{f[w]} = \texttt{()} = v$. Pick $j = 0$ and $v' = \texttt{()}$ and we get that $v = \texttt{()} = v'^{f[w]}$. Clearly $\Gamma \vdash v' : \tau$. Also, for arbitrary $0 \leq k$, $v'^{f[k-j]} = \texttt{()} = e^{f[k]}$, and thus $v'^{f[k-j]} \leq_{\texttt{unit}} e^{f[k]}$.

Case for $e = n$
In this situation, $e^{f[w]} = n = v$. Pick $j = 0$ and $v' = n$ and we get that $v = n = v'^{f[w]}$. Clearly $\Gamma \vdash v' : \tau$. Also, for arbitrary $0 \leq k$, $v'^{f[k-j]} = n = e^{f[k]}$, and thus $v'^{f[k-j]} \leq_{\texttt{int}} e^{f[k]}$.

Case for $e = \langle v_1, v_2 \rangle$
Since $e^{f[w]} \mapsto^0 v$, $e^{f[w]} = v$. Pick $j = 0$ and $v' = e$, which is a value since $e\ \texttt{val}$ by what we know above. We already know that $\Gamma \vdash v' : \tau$, and clearly $e^{f[w]} = v'^{f[w]} = v$. For $j = 0 \leq k$, since $e = v'$, clearly $e^{f[k]} = v'^{f[k]} = v'^{f[k-j]}$, so $v'^{f[k-j]} \leq_\tau e^{f[k]}$.

Case for $e = \texttt{fun}\ h(x : \tau').e'$
Since $e^{f[w]} \mapsto^0 v$, $e^{f[w]} = v$. Pick $j = 0$ and $v' = e$. We already know that $\Gamma \vdash v' : \tau$. Suppose $j = 0 \leq k$. We just need to show that $v'^{f[k-j]} \leq_\tau e^{f[k]}$, which we know because $e^{f[k]} = v'^{f[k]} = v'^{f[k-j]}$.

Case for $e = \widehat{\texttt{fun}}\ h(x : \tau').e'$
Since $e^{f[w]} \mapsto^0 v$, $e^{f[w]} = v$. Pick $j = 0$ and $v' = e$. We already know that $\Gamma \vdash v' : \tau$. Suppose $j = 0 \leq k$. We just need to show that $v'^{f[k-j]} \leq_\tau e^{f[k]}$, which we know because $e^{f[k]} = v'^{f[k]} = v'^{f[k-j]}$.

Case for $e = \Lambda\alpha.e'$
Since $e^{f[w]} \mapsto^0 v$, $e^{f[w]} = v$. Pick $j = 0$ and $v' = e$. We already know that $\Gamma \vdash v' : \tau$. Suppose $j = 0 \leq k$. We just need to show that $v'^{f[k-j]} \leq_\tau e^{f[k]}$, which we know because $e^{f[k]} = v'^{f[k]} = v'^{f[k-j]}$.

Case for $e = \mathtt{pack}[\tau', e']\,\mathtt{as}\,\exists\alpha.\tau$
Since $e^{f[w]} \mapsto^0 v$, $e^{f[w]} = v$. Pick $j = 0$ and $v' = e$. We already know that $\Gamma \vdash v' : \tau$. Suppose $j = 0 \leq k$. We just need to show that $v'^{f[k-j]} \leq_\tau e^{f[k]}$, which we know because $e^{f[k]} = v'^{f[k]} = v'^{f[k-j]}$.

Case for $i > 0$

Case for $e = e_1\,\mathtt{p}\,e_2$
This is where $e^{f[w]} = e_1^{f[w]}\,\mathtt{p}\,e_2^{f[w]} \mapsto^i v$. By *Tintop*, we know that $\Gamma \vdash e_1 : \mathtt{int}$ and $\Gamma \vdash e_2 : \mathtt{int}$. By our outer induction (on the length of the evaluation) on $\Gamma \vdash e_1 : \mathtt{int}$ and $e_1^{f[w]} \mapsto^{i_1} v_1$, since $i_1 < i$, we get that there exist j_1, v_1' such that $\Gamma \vdash v_1' : \mathtt{int}$, $v_1 = v_1'^{f[w]}$, and for all $j_1 \leq k$, $v_1'^{f[k-j_1]} \leq_{\mathtt{int}} e_1^{f[k]}$. Similarly, we get by induction on $\Gamma \vdash e_2 : \mathtt{int}$ and $e_2^{f[w]} \mapsto^{i_2} v_2$, we get that there exist j_2, v_2' such that $\Gamma \vdash v_2' : \mathtt{int}$, $v_2 = v_2'^{f[w]}$, and for all $j_2 \leq k$, $v_2'^{f[k-j_2]} \leq_{\mathtt{int}} e_2^{f[k]}$.

However, since $v_1 : \mathtt{int}$, we know $v_1 = n_1 = v_1'^{f[w]} = v_1'$ for some n_1. Similarly, for some n_2, we get that $v_2 = n_2 = v_2'^{f[w]} = v_2'$. This means that for $j_1 \leq k$, $n_1 \leq_{\mathtt{int}} e_1^{f[k]}$ and for $j_2 \leq k$, $n_2 \leq_{\mathtt{int}} e_2^{f[k]}$. By *Eintop*$_3$, we get that $n_1\,\mathtt{p}\,n_2 \mapsto n$, where $n_1\,\mathtt{p}\,n_2 = n$. Now pick $j = j_1 + j_2$ and $v' = n$. We already know that $\Gamma \vdash v' : \mathtt{int}$, since $n : \mathtt{int}$. It is also clear that $v = n = v'^{f[w]}$. Suppose $j \leq k$, we want to show that $v'^{f[k-j]} \leq_{\mathtt{int}} e^{f[k]}$. But $v'^{f[k-j]} = n$, so we just need to show that $n \leq_{\mathtt{int}} e^{f[k]}$.

Since $n_1\,\mathtt{p}\,n_2 \mapsto n$, by Lemma 5.7 we get that $n \leq_{\mathtt{int}} n_1\,\mathtt{p}\,n_2$. Now define the context $\mathcal{C}_2 = o\,\mathtt{p}\,n_2 : (\cdot \triangleright \mathtt{int}) \rightsquigarrow (\cdot \triangleright \mathtt{int})$. Then since $n_1 \leq_{\mathtt{int}} e_1^{f[k]}$ because $j_1 \leq j \leq k$, we can apply Lemma 5.5 to get that

$$n_1\,\mathtt{p}\,n_2 \leq_{\mathtt{int}} e_1^{f[k]}\,\mathtt{p}\,n_2$$

We can similarly apply Lemma 5.5 to the fact that $n_2 \leq_{\mathtt{int}} e_2^{f[k]}$ since $j_2 \leq j \leq k$, using the context $\mathcal{C}_1 = \mathcal{C}\{e_1^{f[k]}\,\mathtt{p}\,o\} : (\cdot \triangleright \mathtt{int}) \rightsquigarrow (\cdot \triangleright \mathtt{int})$ to get that

$$e_1^{f[k]}\,\mathtt{p}\,n_2 \leq_{\mathtt{int}} e_1^{f[k]}\,\mathtt{p}\,e_2^{f[k]}$$

So by transitivity, we get that

$$n_1\,\mathtt{p}\,n_2 \leq_{\mathtt{int}} e_1^{f[k]}\,\mathtt{p}\,e_2^{f[k]}$$

which is equivalent to

$$(n_1\,\mathtt{p}\,n_2)^{f[k-j]} \leq_{\mathtt{int}} (e_1\,\mathtt{p}\,e_2)^{f[k]}$$

as desired.

Case for $e = \mathtt{ifz}(e_1, e_2, e_3)$
This is where $e^{f[w]} = \mathtt{ifz}(e_1^{f[w]}, e_2^{f[w]}, e_3^{f[w]}) \mapsto^i v$. By *Tifz*, we know that $\Gamma \vdash e_1 : \mathtt{int}$, $\Gamma \vdash e_2 : \tau$ and $\Gamma \vdash e_3 : \tau$. By induction on the length of the evaluation, we get that for $e_1^{f[w]} \mapsto^{i_1} v_1$, there exist j_1, v_1' such that $\Gamma \vdash v_1' : \mathtt{int}$, $v_1 = v_1'^{f[w]}$, and for all $j_1 \leq k$, $v_1'^{f[k-j_1]} \leq_{\mathtt{int}} e_1^{f[k]}$. Let k be such that $j_1 \leq k$, and define the context

$$\mathcal{C}_1 = \mathtt{ifz}(o, e_2^{f[k]}, e_3^{f[k]}) : (\cdot \triangleright \mathtt{int}) \rightsquigarrow (\cdot \triangleright \tau)$$

Using this context to apply Lemma 5.5 to $v_1'^{f[k-j_1]} \leq_{\texttt{int}} e_1^{f[k]}$ from above, we get that

$$\texttt{ifz}(v_1'^{f[k-j_1]}, e_2^{f[k]}, e_3^{f[k]}) \leq_\tau e^{f[k]}$$

Now, since $v_1 : \texttt{int}$, we can case on its value. There are two cases that we consider: $v_1 = 0$, and $v_1 \neq 0$.

If $v_1 = 0$, then we know that $\texttt{ifz}(v_1, e_2^{f[w]}, e_3^{f[w]}) \mapsto e_2^{f[w]}$ by rule $Eifz_2$. We can then induct on the evaluation $e_2^{f[w]} \mapsto^{i_2} v_2 = v$, which tells us that there exist j_2, v_2' such that $\Gamma \vdash v_2' : \tau$, $v_2 = v_2'^{f[w]}$, and for all k such that $j_2 \leq k$, $v_2'^{f[k-j_2]} \leq_\tau e_2^{f[k]}$.

Pick $v' = v_2'$ and $j = j_1 + j_2$. We already know that $\Gamma \vdash v' : \tau$, and since $v = v_2$, we have that $v = v_2 = v_2'^{f[w]} = v'^{f[w]}$. Now suppose we have k such that $j = j_1 + j_2 \leq k$. Defining $k_2 = k - j_1$, we know by our application of induction above that $v_2'^{f[k_2-j_2]} \leq_\tau e_2^{f[k_2]}$. This is equivalent to

$$v_2'^{f[k-j]} \leq_\tau e_2^{f[k-j_1]}$$

However, by Lemma 6.4, we know that $e_2^{f[k-j_1]} \leq_\tau e_2^{f[k]}$, so by the transitivity of contextual approximation we have that

$$v_2'^{f[k-j]} \leq_\tau e_2^{f[k]}$$

We know that $v_1' = 0$ since $v_1 = 0$, so $v_1'^{f[k-j_1]} = 0$. Thus

$$\texttt{ifz}(v_1'^{f[k-j_1]}, e_2^{f[k]}, e_3^{f[k]}) \mapsto e_2^{f[k]}$$

and so by Lemma 5.7 we know that

$$e_2^{f[k]} \leq_\tau \texttt{ifz}(v_1'^{f[k-j_1]}, e_2^{f[k]}, e_3^{f[k]})$$

Now by transitivity, since we know that

$$v_2'^{f[k-j]} \leq_\tau e_2^{f[k]} \leq_\tau \texttt{ifz}(v_1'^{f[k-j_1]}, e_2^{f[k]}, e_3^{f[k]}) \leq_\tau e^{f[k]}$$

we therefore know that $v_2'^{f[k-j]} \leq_\tau e^{f[k]}$, which is what we wanted to show.

If $v_1 \neq 0$, then we know that $\texttt{ifz}(v_1, e_2^{f[w]}, e_3^{f[w]}) \mapsto e_3^{f[w]}$ by rule $Eifz_3$. We can then induct on the evaluation $e_3^{f[w]} \mapsto^{i_3} v_3 = v$, which tells us that there exist j_3, v_3' such that $\Gamma \vdash v_3' : \tau$, $v_3 = v_3'^{f[w]}$, and for all k such that $j_3 \leq k$, $v_3'^{f[k-j_3]} \leq_\tau e_3^{f[k]}$.

Pick $v' = v_3'$ and $j = j_1 + j_3$. We already know that $\Gamma \vdash v' : \tau$, and since $v = v_3$, we have that $v = v_3 = v_3'^{f[w]} = v'^{f[w]}$. Now suppose we have k such that $j = j_1 + j_3 \leq k$. Defining $k_3 = k - j_1$, we know by our application of induction above that $v_3'^{f[k_3-j_3]} \leq_\tau e_3^{f[k_3]}$. This is equivalent to

$$v_3'^{f[k-j]} \leq_\tau e_3^{f[k-j_1]}$$

However, by Lemma 6.4, we know that $e_3^{f[k-j_1]} \leq_\tau e_3^{f[k]}$, so by the transitivity of contextual approximation we have that

$$v_3'^{f[k-j]} \leq_\tau e_3^{f[k]}$$

We know that $v_1' \neq 0$ since $v_1 \neq 0$, so $v_1'^{f[k-j_1]} = 0$. Thus

$$\texttt{ifz}(v_1'^{f[k-j_1]}, e_2^{f[k]}, e_3^{f[k]}) \mapsto e_3^{f[k]}$$

and so by Lemma 5.7 we know that

$$e_3^{f[k]} \leq_\tau \texttt{ifz}(v_1'^{f[k-j_1]}, e_2^{f[k]}, e_3^{f[k]})$$

Now by transitivity, since we know that

$$v_3'^{f[k-j]} \leq_\tau e_3^{f[k]} \leq_\tau \texttt{ifz}(v_1'^{f[k-j_1]}, e_2^{f[k]}, e_3^{f[k]}) \leq_\tau e^{f[k]}$$

we therefore know that $v_3'^{f[k-j]} \leq_\tau e^{f[k]}$, which is what we wanted to show.

Case for $e = e_1\ e_2$
This is where $e^{f[w]} = e_1^{f[w]}\ e_2^{f[w]} \mapsto^i v$. By $Tapp$, we know that $\Gamma \vdash e_1 : \tau' \to \tau$ and $\Gamma \vdash e_2 : \tau'$. By our outer induction (on the length of the evaluation) on $\Gamma \vdash e_1 : \tau' \to \tau$ and $e_1^{f[w]} \mapsto^{i_1} v_1$, since $i_1 < i$, we get that there exist j_1, v_1' such that $\Gamma \vdash v_1' : \tau' \to \tau$, $v_1 = v_1'^{f[w]}$, and for all $j_1 \leq k$, $v_1'^{f[k-j_1]} \leq_{\tau \to \tau'} e_1^{f[k]}$. Similarly, we get by induction on $\Gamma \vdash e_2 : \tau'$ and $e_2^{f[w]} \mapsto^{i_2} v_2$, we get that there exist j_2, v_2' such that $\Gamma \vdash v_2' : \tau'$, $v_2 = v_2'^{f[w]}$, and for all $j_2 \leq k$, $v_2'^{f[k-j_2]} \leq_{\tau'} e_2^{f[k]}$.

We know that $\Gamma \vdash v_1' : \tau' \to \tau$ and $\Gamma \vdash v_2' : \tau'$, so we can apply $Tapp$ to get that $\Gamma \vdash v_1'\ v_2' : \tau$.

Now it would be nice if we could apply induction again to $v_1'\ v_2'$, however we can't, since it's possible for both $j_1 = 0$ and $j_2 = 0$, which means the evaluation isn't actually any smaller. However, we still need to show something similar, and for this we will consider two subcases, making use of an arbitrary k in them:

Consider the case where $v_1' = \texttt{fun}\ h(x : \tau').e'$. Then $(v_1'\ v_2')^{f[k]} \mapsto ([v_1'/h][v_2'/x]e')^{f[k]}$. Now we can apply the outer inductive hypothesis on $([v_1'/h][v_2'/x]e')^{f[w]} \mapsto^{j_3} v_3$, which tells us that there exist j_3, v_3' such that $\Gamma \vdash v_3' : \tau$, $v_3 = v_3'^{f[w]}$, and for all $j_3 \leq k$, $v_3'^{f[k-j_3]} \leq_\tau ([v_1'/h][v_2'/x]e')^{f[k]}$. However for $j_3 \leq k$, since $(v_1'\ v_2')^{f[k]} \mapsto ([v_1'/h][v_2'/x]e')^{f[k]}$, by Lemma 5.7 $([v_1'/h][v_2'/x]e')^{f[k]} \leq_\tau (v_1'\ v_2')^{f[k]}$, and thus by transitivity

$$v_3'^{f[k-j_3]} \leq_\tau (v_1'\ v_2')^{f[k]}$$

The other possible case is where $v_1' = w$, if situation (1) is true. Then $v_1'^{f[k]} = \texttt{fun}^k\ g(x : \tau').e'$. We can then use our outer inductive hypothesis on $([w/g][v_2'/x]e')^{f[w]} \mapsto v_3$ to get that there exist j_3', v_3' such that $\Gamma \vdash v_3' : \tau$, $v_3 = v_3'^{f[w]}$, and for all $j_3' \leq k$, $v_3'^{f[k-j_3']} \leq_\tau ([w/g][v_2'/x]e')^{f[k]}$. Letting $j_3' \leq k$, we know that $([w/g][v_2'/x]e')^{f[k]} = [w^{f[k]}/g][v_2'^{f[k]}/x]e'$ since e' does not have w bound in it. This is equivalent to $[f^k/g][v_2'^{f[k]}/x]e'$, and we know that

$$(\texttt{fun}^{k+1}\ g(x : \tau').e')\ v_2'^{f[k]} \mapsto [f^k/g][v_2'^{f[k]}/x]e'$$

and equivalently

$$v_1'^{f[k+1]}\ v_2'^{f[k]} \mapsto [f^k/g][v_2'^{f[k]}/x]e'$$

so clearly by Lemma 5.7,

$$[f^k/g][v_2'^{f[k]}/x]e' \leq_\tau v_1'^{f[k+1]}\ v_2'^{f[k]}$$

By transitivity, we then get that $v_3'^{f[k-j_3']} \leq_\tau v_1'^{f[k+1]}\ v_2'^{f[k]}$. Since $v_2'^{f[k]} \leq_{\tau'} v_2'^{f[k+1]}$ by Lemma 6.4, we can apply Lemma 5.5 with the context $\mathcal{C}_3 = v_1'^{f[k+1]\frown} o : (\cdot \triangleright \tau' \Rightarrow \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$ to get that

$$v_1'^{f[k+1]}\ v_2'^{f[k]} \leq_\tau v_1'^{f[k+1]}\ v_2'^{f[k+1]}$$

Therefore, by applying transitivity once again, we get that

$$v_3'^{f[k-j_3']} \leq_\tau v_1'^{f[k+1]}\ v_2'^{f[k+1]}$$

If we then define $j_3 = j_3' + 1$, since k was arbitrary we get that for all $j_3 \leq k$,

$$v_3'^{f[k-j_3]} \leq_\tau (v_1'\ v_2')^{f[k]}$$

Now in either of the above cases, we have shown that for all k greater than the corresponding j_3 for the case,

$$v_3'^{f[k-j_3]} \leq_\tau (v_1'\ v_2')^{f[k]}$$

Pick $j = j_1 + j_2 + j_3$ and $v' = v_3'$. We already know that $\Gamma \vdash v' : \tau$. Also, by the combined evaluations above, we know that $v = v_3 = v_3'^{f[w]} = v'^{f[w]}$.

Let $j \leq k$, and thus k is greater than j_1, j_2, j_3. Define $k_3 = k - j_1 - j_2$, and so $j_3 \leq k_3$ by definition of j. Then we know that

$$v_3'^{f[k_3-j_3]} \leq_\tau (v_1'\ v_2')^{f[k_3]}$$

which is equivalent to

$$v_3'^{f[k-j]} \leq_\tau (v_1')^{f[k-j_1-j_2]}\ (v_2')^{f[k-j_1-j_2]}$$

Now define $k_1 = k - j_2$, clearly $j_1 \leq k_1$. Using what we know from induction as stated above, we get that

$$v_1'^{f[k_1-j_1]} \leq_{\tau' \to \tau} e_1^{f[k_1]}$$

which is equivalent to

$$v_1'^{f[k-j_1-j_2]} \leq_{\tau' \to \tau} e_1^{f[k-j_2]}$$

By Lemma 6.4, we get know that $e_1^{f[k-j_2]} \leq_{\tau' \to \tau} e_1^{f[k]}$, so by the transitivity of contextual approximation,

$$v_1'^{f[k-j_1-j_2]} \leq_{\tau' \to \tau} e_1^{f[k]}$$

Similarly defining $k_2 = k - j_1$, by induction we get that

$$v_2'^{f[k_2-j_2]} \leq_{\tau'} e_2^{f[k_2]}$$

which is equivalent to

$$v_2'^{f[k-j_1-j_2]} \leq_{\tau'} e_2^{f[k-j_1]}$$

By Lemma 6.4, we get know that $e_2^{f[k-j_1]} \leq_{\tau'} e_2^{f[k]}$, so by transitivity,

$$v_2'^{f[k-j_1-j_2]} \leq_{\tau'} e_2^{f[k]}$$

We can apply Lemma 5.5 to the context

$$\mathcal{C}_2 = o\ v_2'^{f[k-j_1-j_2]} : (\cdot \triangleright \tau_1 \to \tau_2) \rightsquigarrow (\cdot \triangleright \texttt{int})$$

and $v_1'^{f[k-j_1-j_2]} \leq_{\tau' \to \tau} e_1^{f[k]}$ to get that

$$v_1'^{f[k-j_1-j_2]}\ v_2'^{f[k-j_1-j_2]} \leq_\tau e_1^{f[k]}\ v_2'^{f[k-j_1-j_2]}$$

Similarly, we can apply Lemma 5.5 to the context

$$\mathcal{C}_1 = e_1^{f[k]}\ o : (\cdot \triangleright \tau_1 \to \tau_2) \rightsquigarrow (\cdot \triangleright \texttt{int})$$

and $v_2'^{f[k-j_1-j_2]} \leq_{\tau'} e_2^{f[k]}$ to get that

$$e_1^{f[k]}\ v_2'^{f[k-j_1-j_2]} \leq_\tau e_1^{f[k]}\ e_2^{f[k]}$$

Then we can apply the transitivity of contextual approximation to

$$v_3'^{f[k-j]} \leq_\tau v_1'^{f[k-j_1-j_2]}\ v_2'^{f[k-j_1-j_2]} \leq_\tau e_1^{f[k]}\ v_2'^{f[k-j_1-j_2]} \leq_\tau e_1^{f[k]}\ e_2^{f[k]}$$

which gets us that

$$v_3'^{f[k-j]} \leq_\tau (e_1\ e_2)^{f[k]}$$

which is what we wanted to show. Since k was arbitrary, we have shown this to be true for all $j \leq k$, as desired.

Case for $e = e_1 \frown e_2$
This is where $e^{f[w]} = e_1^{f[w]} \frown e_2^{f[w]} \mapsto^i v$. By *Tccapp*, we know that $\Gamma \vdash e_1 : \tau' \Rightarrow \tau$ and $\Gamma \vdash e_2 : \tau'$. By our outer induction (on the length of the evaluation) on $\Gamma \vdash e_1 : \tau' \Rightarrow \tau$ and $e_1^{f[w]} \mapsto^{i_1} v_1$, since $i_1 < i$, we get that there exist j_1, v_1' such that $\Gamma \vdash v_1' : \tau' \Rightarrow \tau$, $v_1 = v_1'^{f[w]}$, and for all $j_1 \leq k$, $v_1'^{f[k-j_1]} \leq_{\tau' \Rightarrow \tau} e_1^{f[k]}$. Similarly, we get by induction on $\Gamma \vdash e_2 : \tau'$ and $e_2^{f[w]} \mapsto^{i_2} v_2$, we get that there exist j_2, v_2' such that $\Gamma \vdash v_2' : \tau'$, $v_2 = v_2'^{f[w]}$, and for all $j_2 \leq k$, $v_2'^{f[k-j_2]} \leq_{\tau'} e_2^{f[k]}$.

We know that $\Gamma \vdash v_1' : \tau' \Rightarrow \tau$ and $\Gamma \vdash v_2' : \tau'$, so we can apply *Tccapp* to get that $\Gamma \vdash v_1' \frown v_2' : \tau$.

Now it would be nice if we could apply induction again to $v_1' \frown v_2'$, however we can't, since it's possible for both $j_1 = 0$ and $j_2 = 0$, which means the evaluation isn't actually any smaller. However, we still need to show something similar, and for this we will consider two subcases, making use of an arbitrary k in them:

> Consider the case where $v_1' = \widehat{\texttt{fun}}\ h(x : \tau').e'$. Then $(v_1' \frown v_2')^{f[k]} \mapsto ([v_1'/h][v_2'/x]e')^{f[k]}$. Now we can apply the outer inductive hypothesis on $([v_1'/h][v_2'/x]e')^{f[w]} \mapsto^{j_3} v_3$, which tells us that there exist j_3, v_3' such that $\Gamma \vdash v_3' : \tau$, $v_3 = v_3'^{f[w]}$, and for all $j_3 \leq k$, $v_3'^{f[k-j_3]} \leq_\tau ([v_1'/h][v_2'/x]e')^{f[k]}$. However for $j_3 \leq k$, since $(v_1' \frown v_2')^{f[k]} \mapsto ([v_1'/h][v_2'/x]e')^{f[k]}$, by Lemma 5.7 $([v_1'/h][v_2'/x]e')^{f[k]} \leq_\tau (v_1' \frown v_2')^{f[k]}$, and thus by transitivity
>
> $$v_3'^{f[k-j_3]} \leq_\tau (v_1' \frown v_2')^{f[k]}$$
>
> The other possible case is where $v_1' = w$, if situation (2) is true. Then $v_1'^{f[k]} = \widehat{\texttt{fun}}^k\ g(x : \tau').e'$. We can then use our outer inductive hypothesis on $([w/g][v_2'/x]e')^{f[w]} \mapsto v_3$ to get that there exist j_3', v_3' such that $\Gamma \vdash v_3' : \tau$, $v_3 = v_3'^{f[w]}$, and for all $j_3' \leq k$, $v_3'^{f[k-j_3']} \leq_\tau ([w/g][v_2'/x]e')^{f[k]}$. Letting $j_3' \leq k$, we know that $([w/g][v_2'/x]e')^{f[k]} =$

$[w^{f[k]}/g][v_2'^{f[k]}/x]e'$ since e' does not have w bound in it. This is equivalent to $[f^k/g][v_2'^{f[k]}/x]e'$, and we know that

$$(\texttt{fun}^{k+1}\ g(x:\tau').e')^\frown v_2'^{f[k]} \mapsto [f^k/g][v_2'^{f[k]}/x]e'$$

and equivalently

$$v_1'^{f[k+1]\frown} v_2'^{f[k]} \mapsto [f^k/g][v_2'^{f[k]}/x]e'$$

so clearly by Lemma 5.7,

$$[f^k/g][v_2'^{f[k]}/x]e' \leq_\tau v_1'^{f[k+1]\frown} v_2'^{f[k]}$$

By transitivity, we then get that $v_3'^{f[k-j_3']} \leq_\tau v_1'^{f[k+1]\frown} v_2'^{f[k]}$. Since $v_2'^{f[k]} \leq_{\tau'} v_2'^{f[k+1]}$ by Lemma 6.4, we can apply Lemma 5.5 with the context $\mathcal{C}_3 = \mathcal{C}\{v_1'^{f[k+1]\frown} o\} : (\cdot \triangleright \tau' \Rightarrow \tau) \rightsquigarrow (\cdot \triangleright \texttt{int})$ to get that

$$v_1'^{f[k+1]\frown} v_2'^{f[k]} \leq_\tau v_1'^{f[k+1]\frown} v_2'^{f[k+1]}$$

Therefore, by applying transitivity once again, we get that

$$v_3'^{f[k-j_3']} \leq_\tau v_1'^{f[k+1]\frown} v_2'^{f[k+1]}$$

If we then define $j_3 = j_3' + 1$, since k was arbitrary we get that for all $j_3 \leq k$,

$$v_3'^{f[k-j_3]} \leq_\tau (v_1'^\frown v_2')^{f[k]}$$

Now in either of the above cases, we have shown that for all k greater than the corresponding j_3 for the case,

$$v_3'^{f[k-j_3]} \leq_\tau (v_1'^\frown v_2')^{f[k]}$$

Pick $j = j_1 + j_2 + j_3$ and $v' = v_3'$. We already know that $\Gamma \vdash v' : \tau$. Also, by the combined evaluations above, we know that $v = v_3 = v_3'^{f[w]} = v'^{f[w]}$.

Let $j \leq k$, and thus k is greater than j_1, j_2, j_3. Define $k_3 = k - j_1 - j_2$, and so $j_3 \leq k_3$ by definition of j. Then we know that

$$v_3'^{f[k_3-j_3]} \leq_\tau (v_1'^\frown v_2')^{f[k_3]}$$

which is equivalent to

$$v_3'^{f[k-j]} \leq_\tau (v_1')^{f[k-j_1-j_2]\frown}(v_2')^{f[k-j_1-j_2]}$$

Now define $k_1 = k - j_2$, clearly $j_1 \leq k_1$. Using what we know from induction as stated above, we get that

$$v_1'^{f[k_1-j_1]} \leq_{\tau'\Rightarrow\tau} e_1^{f[k_1]}$$

which is equivalent to

$$v_1'^{f[k-j_1-j_2]} \leq_{\tau'\Rightarrow\tau} e_1^{f[k-j_2]}$$

By Lemma 6.4, we get know that $e_1^{f[k-j_2]} \leq_{\tau'\Rightarrow\tau} e_1^{f[k]}$, so by the transitivity of contextual approximation,

$$v_1'^{f[k-j_1-j_2]} \leq_{\tau'\Rightarrow\tau} e_1^{f[k]}$$

Similarly defining $k_2 = k - j_1$, by induction we get that

$$v_2'^{f[k_2-j_2]} \leq_{\tau'} e_2^{f[k_2]}$$

which is equivalent to

$$v_2'^{f[k-j_1-j_2]} \leq_{\tau'} e_2^{f[k-j_1]}$$

By Lemma 6.4, we get know that $e_2^{f[k-j_1]} \leq_{\tau'} e_2^{f[k]}$, so by transitivity,

$$v_2'^{f[k-j_1-j_2]} \leq_{\tau'} e_2^{f[k]}$$

We can apply Lemma 5.5 to the context

$$\mathcal{C}_2 = o\frown v_2'^{f[k-j_1-j_2]} : (\cdot \triangleright \tau_1 \Rightarrow \tau_2) \rightsquigarrow (\cdot \triangleright \texttt{int})$$

and $v_1'^{f[k-j_1-j_2]} \leq_{\tau' \Rightarrow \tau} e_1^{f[k]}$ to get that

$$v_1'^{f[k-j_1-j_2]}\frown v_2'^{f[k-j_1-j_2]} \leq_{\tau} e_1^{f[k]}\frown v_2'^{f[k-j_1-j_2]}$$

Similarly, we can apply Lemma 5.5 to the context

$$\mathcal{C}_1 = e_1^{f[k]}\frown o : (\cdot \triangleright \tau_1 \Rightarrow \tau_2) \rightsquigarrow (\cdot \triangleright \texttt{int})$$

and $v_2'^{f[k-j_1-j_2]} \leq_{\tau'} e_2^{f[k]}$ to get that

$$e_1^{f[k]}\frown v_2'^{f[k-j_1-j_2]} \leq_{\tau} e_1^{f[k]}\frown e_2^{f[k]}$$

Then we can apply the transitivity of contextual approximation to

$$v_3'^{f[k-j]} \leq_{\tau} v_1'^{f[k-j_1-j_2]}\frown v_2'^{f[k-j_1-j_2]} \leq_{\tau} e_1^{f[k]}\frown v_2'^{f[k-j_1-j_2]} \leq_{\tau} e_1^{f[k]}\frown e_2^{f[k]}$$

which gets us that

$$v_3'^{f[k-j]} \leq_{\tau} (e_1\frown e_2)^{f[k]}$$

which is what we wanted to show. Since k was arbitrary, we have shown this to be true for all $j \leq k$, as desired.

Case for $e = \langle e_1, e_2 \rangle$
This is where $e^{f[w]} = \langle e_1, e_2 \rangle^{f[w]} \mapsto^i v$. By $Tpair$, we know that $\Gamma \vdash e_1 : \tau_1$ and $\Gamma \vdash e_2 : \tau_2$. We can apply our inner induction on the size of the term to $e_1^{f[w]} \mapsto^{i_1} v_1$, which we must do since the evaluation may not be any shorter (meaning $i = i_1$). From this we get that there exist j_1, v_1' such that $\Gamma \vdash v_1' : \tau_1$, $v_1 = v_1'^{f[w]}$, and for all $k \geq j_1$, $v_1'^{f[k-j_1]} \leq_{\tau_1} e_1^{f[k]}$. Similarly, we can apply our inner induction on the size of the term to $e_2^{f[w]} \mapsto^{i_2} v_2$ For the same reason as before). From this we get that there exist j_2, v_2' such that $\Gamma \vdash v_2' : \tau_2$, $v_2 = v_2'^{f[w]}$, and for all $k \geq j_2$, $v_2'^{f[k-j_2]} \leq_{\tau_2} e_2^{f[k]}$.

Pick $j = j_1 + j_2$ and $v' = \langle v_1', v_2' \rangle$. By rule $Tpair$, we know that $\Gamma \vdash v' : \tau_1 \times \tau_2$. Also, since $v_1 = v_1'^{f[w]}$ and $v_2 = v_2'^{f[w]}$, we know that $v = \langle v_1, v_2 \rangle = \langle v_1'^{f[w]}, v_2'^{f[w]} \rangle = \langle v_1', v_2' \rangle^{f[w]} = v'^{f[w]}$. Let $j \leq k$, and define $k_1 = k - j_2$ and $k_2 = k - j_1$. Since $j_1 \leq k_1$, by the above we know that $v_1'^{f[k_1-j_1]} \leq_{\tau_1} e_1^{f[k_1]}$. This is equivalent to

$$v_1'^{f[k-j]} \leq_{\tau_1} e_1^{f[k-j_2]}$$

By Lemma 6.4, we know that $e_1^{f[k-j_2]} \leq_{\tau_1} e_1^{f[k]}$, so by the transitivity of contextual approximation,

$$v_1'^{f[k-j]} \leq_{\tau_1} e_1^{f[k]}$$

Similarly, using k_2 in the same way as k_1 we can deduce that

$$v_2'^{f[k-j]} \leq_{\tau_2} e_2^{f[k]}$$

again using our results from induction and Lemma 6.4.

Using the context $\mathcal{C}_1 = \langle o, v_2'^{f[k-j]} \rangle : (\cdot \triangleright \tau_1) \rightsquigarrow (\cdot \triangleright \tau_1 \times \tau_2)$ we can apply Lemma 5.5 to

$$v_1'^{f[k-j]} \leq_{\tau_1} e_1^{f[k]}$$

which gets us that

$$\langle v_1'^{f[k-j]}, v_2'^{f[k-j]} \rangle \leq_{\tau_1 \times \tau_2} \langle e_1^{f[k]}, v_2'^{f[k-j]} \rangle$$

Using the context $\mathcal{C}_2 = \langle e_1^{f[k]}, o \rangle : (\cdot \triangleright \tau_2) \rightsquigarrow (\cdot \triangleright \tau_1 \times \tau_2)$ we can apply Lemma 5.5 to

$$v_2'^{f[k-j]} \leq_{\tau_2} e_2^{f[k]}$$

which gets us that

$$\langle e_1^{f[k]}, v_2'^{f[k-j]} \rangle \leq_{\tau_1 \times \tau_2} \langle e_1^{f[k]}, e_2^{f[k]} \rangle$$

Then by transitivity of contextual approximation, we have that

$$\langle v_1'^{f[k-j]}, v_2'^{f[k-j]} \rangle \leq_{\tau_1 \times \tau_2} \langle e_1^{f[k]}, e_2^{f[k]} \rangle$$

Or equivalently, $v'^{f[k-j]} \leq_{\tau_1 \times \tau_2} e^{f[k]}$, as desired.

Case for $e = \pi_i e'$
This is where $e^{f[w]} = \pi_i e'^{f[w]} \mapsto^i v$. By $Tproj$, we know that $\Gamma \vdash e' : \tau_1 \times \tau_2$, where $\tau_i = \tau$. By induction on $e'^{f[w]} \mapsto^{i'} \langle v_1, v_2 \rangle$, which we can do by the length since $i' < i$, we get that there exist j', v'' such that $\Gamma \vdash v'' : \tau_1 \times \tau_2$, $v = v''^{f[w]}$, and for all $j' \leq k$, $v''^{f[k-j']} \leq_{\tau_1 \times \tau_2} e'^{f[k]}$.

Since $\Gamma \vdash v'' : \tau_1 \times \tau_2$, by rule $Tpair$ we know that $v'' = \langle v_1', v_2' \rangle$ for some v_1', v_2' such that $\Gamma \vdash v_1' : \tau_1$ and $\Gamma \vdash v_2' : \tau_2$. Pick $j = j'$ and $v' = v_i'$. By above we know that $\Gamma \vdash v_i' : \tau$, and since $\langle v_1, v_2 \rangle = v''^{f[w]}$, $v = v_i = v_i'^{f[w]}$. Suppose $j = j' \leq k$. Since $v''^{f[k-j]} \leq_{\tau_1 \times \tau_2} e'^{f[k]}$, We can apply Lemma 5.5 to the context $\mathcal{C}' = \mathcal{C}\{\pi_i o\} : (\cdot \triangleright \tau_1 \times \tau_2) \rightsquigarrow (\cdot \triangleright \tau)$ to get that

$$\pi_i \langle v_1'^{f[k-j]}, v_2'^{f[k-j]} \rangle \leq_\tau \pi_i e'^{f[k]}$$

Then, using Lemma 5.7 on the fact that $\pi_i \langle v_1'^{f[k-j]}, v_2'^{f[k-j]} \rangle \mapsto v_i'^{f[k-j]}$, we can get that and so $v_i'^{f[k-j]} \leq_\tau \pi_i e'^{f[k]}$ as desired.

Case for $e = e'[\tau']$
This is where $e^{f[w]} = (e'[\tau'])^{f[w]} \mapsto^i v$. By $Ttapp$, we know that $\Gamma \vdash e' : \forall \alpha.\tau$. By induction on $e'^{f[w]} \mapsto^{i_1} v_1$, which we can do by length since $i_1 < i$, we get that there exist j_1, v_1' such that $\Gamma \vdash v_1' : \forall \alpha.\tau$, $v_1 = v_1'^{f[w]}$, and for all $j_1 \leq k$, $v_1'^{f[k-j_1]} \leq_{\forall \alpha.\tau} e'^{f[k]}$.

Since $\Gamma \vdash v_1' : \forall \alpha.\tau$, we know by rule $Ttlam$ that $v_1' = \Lambda \alpha.e''$ for some e'' such that $\alpha; \Gamma \vdash e'' : \tau$. By rule $Etapp_2$ we know that $((\Lambda \alpha.e'')[\tau'])^{f[w]} \mapsto ([\tau'/\alpha]e'')^{f[w]}$. Now we can again apply induction on the evaluation $([\tau'/\alpha]e'')^{f[w]} \mapsto^{i_2} v_2$ to get that there exist j_2, v_2' such that $\Gamma \vdash v_2' : \tau$, $v_2 = v_2'^{f[w]}$, and for all $j_2 \leq k$, $v_2'^{f[k-j_2]} \leq_\tau ([\tau'/\alpha]e'')^{f[k]}$.

Pick $j = j_1 + j_2$ and $v' = v'_2$. We already know that $\Gamma \vdash v' : \tau$ by the above, and since $v = v_2$, we have that $v = v_2 = {v'_2}^{f[w]} = v'^{f[w]}$. Now suppose we have k such that $j \leq k$.

Defining $k_2 = k - j_1$, by our above induction we have that

$$ {v'_2}^{f[k_2 - j_2]} \leq_\tau ([\tau'/\alpha]e'')^{f[k_2]} $$

which is equivalent to

$$ {v'_2}^{f[k-j]} \leq_\tau ([\tau'/\alpha]e'')^{f[k-j_1]} $$

Since we also know that $(v'_1[\tau'])^{f[k-j_1]} \mapsto ([\tau'/\alpha]e'')^{f[k-j_1]}$, by Lemma 5.7

$$ ([\tau'/\alpha]e'')^{f[k-j_1]} \leq_\tau (v'_1[\tau'])^{f[k-j_1]} $$

Therefore by transitivity we know that

$$ {v'_2}^{f[k-j]} \leq_\tau (v'_1[\tau'])^{f[k-j_1]} $$

Now by our induction results from above, we know that

$$ {v'_1}^{f[k-j_1]} \leq_{\forall\alpha.\tau} e'^{f[k]} $$

Using the context $\mathcal{C}_1 = o[\tau'] : (\cdot \triangleright \forall\alpha.\tau) \rightsquigarrow (\cdot \triangleright \tau)$ we can apply Lemma 5.5 to get that

$$ ({v'_1}^{f[k-j_1]})[\tau'] \leq_\tau (e'^{f[k]})[\tau'] $$

which is equivalent to

$$ (v'_1[\tau'])^{f[k-j_1]} \leq_\tau (e'[\tau'])^{f[k]} $$

Then by transitivity, we have that

$$ {v'_2}^{f[k-j]} \leq_\tau (e'[\tau'])^{f[k]} $$

or equivalently $v'^{f[k-j]} \leq_\tau e^{f[k]}$, which is what we wanted to show.

Case for $e = \texttt{unpack}[\alpha, x] = e_1 \,\texttt{in}\, e_2$
This is where $e^{f[w]} = (\texttt{unpack}[\alpha, x] = e_1 \,\texttt{in}\, e_2)^{f[w]} \mapsto^i v$. By *Tunpack*, we know that $\Gamma \vdash e_1 : \exists\alpha.\tau_1$. By induction on $e_1^{f[w]} \mapsto^{i_1} v_1$, which we can do by length since $i_1 < i$, we get that there exist j_1, v'_1 such that $\Gamma \vdash v'_1 : \exists\alpha.\tau_1$, $v_1 = {v'_1}^{f[w]}$, and for all $j_1 \leq k$, ${v'_1}^{f[k-j_1]} \leq_{\exists\alpha.\tau_1} e'^{f[k]}$.

Since $\Gamma \vdash v'_1 : \exists\alpha.\tau_1$, we know by rule *Tpack* that $v'_1 = \texttt{pack}[\tau', e'_1] \,\texttt{as}\, \exists\alpha.\tau_1$ for some e'_1 such that $\alpha; \Gamma \vdash e'_1 : \tau_1$. By rule *Eunpack*$_2$ we know that

$$ (\texttt{unpack}[\alpha, x] = (\texttt{pack}[\tau', e'_1] \,\texttt{as}\, \exists\alpha.\tau_1) \,\texttt{in}\, e_2)^{f[w]} \mapsto ([\tau'/\alpha][e'_1/x]e_2)^{f[w]} $$

Now we can again apply induction on the evaluation $([\tau'/\alpha][e'_1/x]e_2)^{f[w]} \mapsto^{i_2} v_2$ to get that there exist j_2, v'_2 such that $\Gamma \vdash v'_2 : \tau$, $v_2 = {v'_2}^{f[w]}$, and for all $j_2 \leq k$, ${v'_2}^{f[k-j_2]} \leq_\tau ([\tau'/\alpha][e'_1/x]e_2)^{f[k]}$.

Pick $j = j_1 + j_2$ and $v' = v'_2$. We already know that $\Gamma \vdash v' : \tau$ by the above, and since $v = v_2$, we have that $v = v_2 = {v'_2}^{f[w]} = v'^{f[w]}$. Now suppose we have k such that $j \leq k$.

Defining $k_2 = k - j_1$, by our above induction we have that

$$ {v'_2}^{f[k_2 - j_2]} \leq_\tau ([\tau'/\alpha][e'_1/x]e_2)^{f[k_2]} $$

which is equivalent to

$$v_2'^{f[k-j]} \leq_\tau ([\tau'/\alpha][e_1'/x]e_2)^{f[k-j_1]}$$

Since we also know that

$$(\texttt{unpack}[\alpha, x] = v_1' \,\texttt{in}\, e_2)^{f[k-j_1]} \mapsto ([\tau'/\alpha][e_1'/x]e_2)^{f[k-j_1]}$$

we then know by Lemma 5.7 that

$$([\tau'/\alpha][e_1'/x]e_2)^{f[k-j_1]} \leq_\tau (\texttt{unpack}[\alpha, x] = v_1' \,\texttt{in}\, e_2)^{f[k-j_1]}$$

Therefore by transitivity we know that

$$v_2'^{f[k-j]} \leq_\tau (\texttt{unpack}[\alpha, x] = v_1' \,\texttt{in}\, e_2)^{f[k-j_1]}$$

Now by our induction results from above, we know that

$$v_1'^{f[k-j_1]} \leq_{\exists\alpha.\tau} e_1^{f[k]}$$

Using the context $\mathcal{C}_1 = \texttt{unpack}[\alpha, x] = o \,\texttt{in}\, e_2 : (\cdot \triangleright \exists\alpha.\tau) \rightsquigarrow (\cdot \triangleright \tau)$ we can apply Lemma 5.5 to get that

$$\texttt{unpack}[\alpha, x] = v_1'^{f[k-j_1]} \,\texttt{in}\, e_2 \leq_\tau \texttt{unpack}[\alpha, x] = e_1^{f[k]} \,\texttt{in}\, e_2$$

which is equivalent to

$$(\texttt{unpack}[\alpha, x] = v_1' \,\texttt{in}\, e_2)^{f[k-j_1]} \leq_\tau (\texttt{unpack}[\alpha, x] = e_1 \,\texttt{in}\, e_2)^{f[k]}$$

Then by transitivity using the above, we have that

$$v_2'^{f[k-j]} \leq_\tau (\texttt{unpack}[\alpha, x] = e_1 \,\texttt{in}\, e_2)^{f[k]}$$

or equivalently $v'^{f[k-j]} \leq_\tau e^{f[k]}$, which is what we wanted to show.

□

6.2 Compactness

Theorem 6.6. Given a closed recursive function value f, where $\cdot \vdash f : \tau$ such that either $\tau = \tau_1 \to \tau_2$ or $\tau = \tau_1 \Rightarrow \tau_2$, then for all terms e such that $w : \tau \vdash e : \tau'$, $e^{f[w]} \downarrow \Leftrightarrow \exists n.e^{f[n]} \downarrow$ This can also be expressed as $e^{f[w]} \simeq e^{f[n]}$ for some n.

Proof. First we will show the forward direction, $e^{f[w]} \downarrow \Rightarrow \exists n.e^{f[n]} \downarrow$ Suppose $e^{f[w]} \downarrow$, that is, for some $\cdot \vdash v : \tau'$, $e^{f[w]} \mapsto^* v$. Then we can apply Lemma 6.5 to get that there exist j, v' such that $w : \tau \vdash v' : \tau'$, $v = v'^{f[w]}$, and for all $k \geq j$, $v'^{f[k-j]} \leq_{\tau'} e^{f[k]}$.

Pick $n = j + 1$. For the empty context, we get that $v'^{f[n-j]} \lesssim e^{f[n]}$ or equivalently $v'^{f[1]} \lesssim e^{f[n]}$. Since $v'^{f[1]}$ is a value, we know that $v'^{f[1]} \downarrow$, so by definition then $e^{f[n]} \downarrow$, as desired.

Now we will show the backward direction, $\exists n.e^{f[n]} \downarrow \Rightarrow e^{f[w]} \downarrow$ Assume that for some n, we have that $e^{f[n]} \downarrow$. However, by Corollary 6.3, we know that $f^n \leq_\tau f^w$, so then by Lemma 5.9 we know that $e^{f[n]} \leq_{\tau'} e^{f[w]}$, which gets us the desired result by definition, that $e^{f[w]} \downarrow$. $\square$

7 Relations

Definition The set $\mathtt{Val}(\tau)$ is the set of all values v of type τ.

Definition A relation over values $\mathtt{R}$ is a subset of $\mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$, which consists of all pairs of values of types τ and τ', respectively. We say $v \;\mathtt{R}\; v'$ iff $(v, v') \in \mathtt{R}$, and thus $\cdot \vdash v : \tau$ and $\cdot \vdash v' : \tau'$.

Definition Given a relation $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$ over values, we can convert it to a relation over functions, written $\mathtt{R}^{\mathtt{S}}$, such that $\mathtt{R}^{\mathtt{S}} \subseteq \mathtt{Val}(\tau \to \tau_2) \times \mathtt{Val}(\tau' \to \tau_2')$. These functions can be thought of as "stacks" or continuations that contain remaining computation to be done. We define this continuation relation $\mathtt{R}^{\mathtt{S}}$ as follows:

$$f \;\mathtt{R}^{\mathtt{S}}\; f' \text{ iff } \forall v, v'.\text{ if } v \;\mathtt{R}\; v' \text{ then } f\ v \simeq f'\ v'$$

Similarly, given such a function relation $\mathtt{R} \subseteq \mathtt{Val}(\tau \to \tau_2) \times \mathtt{Val}(\tau' \to \tau_2')$, we can convert it back into a value relation, written $\mathtt{R}^{\mathtt{T}}$, such that $\mathtt{R}^{\mathtt{T}} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$. We define it as follows:

$$v \;\mathtt{R}^{\mathtt{T}}\; v' \text{ iff } \forall f, f'.\text{ if } f \;\mathtt{R}\; f' \text{ then } f\ v \simeq f'\ v'$$

The benefit of these relation transformations is clearer after seeing the properties that they provide. For simplicity, the letter symbols f and g and their variants will be used to indicate functions.

7.1 ST Closure

Definition Given a relation $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$, we call $\mathtt{R}^{\mathtt{ST}}$ the ST-closure of $\mathtt{R}$. If $\mathtt{R} = \mathtt{R}^{\mathtt{ST}}$, then we say that $\mathtt{R}$ is ST-closed.

Lemma 7.1. For a given relation $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$, $\mathtt{R} \subseteq \mathtt{R}^{\mathtt{ST}}$ (meaning that the ST-closure is inflationary).

Proof. Assume $v \;\mathtt{R}\; v'$. We want to show that $v \;\mathtt{R}^{\mathtt{ST}}\; v'$.

By the definition of $\mathtt{R}^{\mathtt{S}}$, we know that if $(f, f') \in \mathtt{R}^{\mathtt{S}}$ then $f\ v \simeq f'\ v'$, since by assumption $v \;\mathtt{R}\; v'$. Then by the definition of $\mathtt{R}^{\mathtt{ST}}$, we have that $v \;\mathtt{R}^{\mathtt{ST}}\; v'$. □

Lemma 7.2. For a given relation $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$, $\mathtt{R}^{\mathtt{ST}} = \mathtt{R}^{\mathtt{STST}}$ (meaning that the ST-closure is idempotent).

Proof. First, we need to show that if $v \;\mathtt{R}^{\mathtt{ST}}\; v'$, then $v \;\mathtt{R}^{\mathtt{STST}}\; v'$. We already know this from the proof that $\mathtt{R} \subseteq \mathtt{R}^{\mathtt{ST}}$, so this case is done.

Now we need to show that if $v \;\mathtt{R}^{\mathtt{STST}}\; v'$, then $v \;\mathtt{R}^{\mathtt{ST}}\; v'$. Assume that $v \;\mathtt{R}^{\mathtt{STST}}\; v'$. This means that if $f \;\mathtt{R}^{\mathtt{STS}}\; f'$, then $f\ v \simeq f'\ v'$. For $v \;\mathtt{R}^{\mathtt{ST}}\; v'$ to be true, we need to show that for all $f \;\mathtt{R}^{\mathtt{S}}\; f'$, $f\ v \simeq f'\ v'$. To show this we just need to prove that $\mathtt{R}^{\mathtt{S}} \subseteq \mathtt{R}^{\mathtt{STS}}$, which can be shown in a manner very similar to the proof of $\mathtt{R} \subseteq \mathtt{R}^{\mathtt{ST}}$ in Lemma 7.1.

Thus, the two sets must be equal. □

Definition We can extend a relation $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$ over values to a relation $\mathtt{R}^{\mathtt{E}}$ over terms by considering only terms that evaluate to those values:

$e\ \mathtt{R}^{\mathtt{E}}\ e'$ iff $e \simeq e'$ and $\forall v, v'$. if $e \mapsto^* v$ and $e' \mapsto^* v'$ then $v\ \mathtt{R}\ v'$

where $\cdot \vdash e : \tau$ and $\cdot \vdash e' : \tau'$.

Corollary 7.3. Given a relation $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$, then $\mathtt{R} \subseteq \mathtt{R}^{\mathtt{E}}$.

Lemma 7.4. Given a relation $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$, if $e_1' \mapsto e_1$ then $(e_1, e_2) \in \mathtt{R}^{\mathtt{E}} \Leftrightarrow (e_1', e_2) \in \mathtt{R}^{\mathtt{E}}$. Also, if $e_2' \mapsto e_2$, then $(e_1, e_2) \in \mathtt{R}^{\mathtt{E}} \Leftrightarrow (e_1, e_2') \in \mathtt{R}^{\mathtt{E}}$

Proof. We will first show the forward direction. Suppose that $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$ where $(e_1, e_2) \in \mathtt{R}^{\mathtt{E}}$ such that $e_1' \mapsto e_1$. We want to show that $(e_1', e_2) \in \mathtt{R}^{\mathtt{E}}$.

First we need to show that $e_1' \simeq e_2$. We know that $e_1 \simeq e_2$ by assumption. We also know that $e_1' \simeq e_1$ due to rule *Hstep*, so therefore $e_1' \simeq e_2$ by transitivity.

Now we just need to show that $\forall v_1, v_2$. if $e_1' \mapsto^* v_1$ and $e_2 \mapsto^* v_2$ then $(v_1, v_2) \in \mathtt{R}$. But if $e_1' \mapsto^* v_1$, then $e_1 \mapsto^* v_1$ since $e_1' \mapsto e_1$. Since we already know that $\forall v_1, v_2$. if $e_1 \mapsto^* v_1$ and $e_2 \mapsto^* v_2$ then $(v_1, v_2) \in \mathtt{R}$ by assumption, the result follows.

For the opposite direction, suppose that $(e_1', e_2) \in \mathtt{R}^{\mathtt{E}}$ and that $e_1' \mapsto e_1$. We want to show that $(e_1, e_2) \in \mathtt{R}^{\mathtt{E}}$.

First we need to show that $e_1 \simeq e_2$. We know that $e_1' \simeq e_2$ by assumption. We also know that $e_1' \simeq e_1$ due to rule *Hstep*, so therefore $e_1' \simeq e_2$ by transitivity.

Now we just need to show that $\forall v_1, v_2$. if $e_1 \mapsto^* v_1$ and $e_2 \mapsto^* v_2$ then $(v_1, v_2) \in \mathtt{R}$. But if $e_1 \mapsto^* v_1$, then $e_1' \mapsto^* v_1$ since $e_1' \mapsto e_1$. Since we already know that $\forall v_1, v_2$. if $e_1' \mapsto^* v_1$ and $e_2 \mapsto^* v_2$ then $(v_1, v_2) \in \mathtt{R}$ by assumption, the result follows.

Proving the other case works out in essentially the same way. □

Corollary 7.5. Given a relation $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$, if $e_1' \mapsto^* e_1$ and $e_2' \mapsto^* e_2$ then

$$(e_1, e_2) \in \mathtt{R}^{\mathtt{E}} \Leftrightarrow (e_1', e_2') \in \mathtt{R}^{\mathtt{E}}$$

Lemma 7.6. Given a relation $\mathtt{R} \subset \mathtt{Val}(\tau) \times \mathtt{Val}(\tau)$, if $(f, f') \in \mathtt{R}^{\mathtt{S}}$ and $(e, e') \in \mathtt{R}^{\mathtt{STE}}$, where $\cdot \vdash e : \tau$ and $\cdot \vdash e' : \tau$, then $f\ e \simeq f'\ e'$.

Proof. Let $(f, f') \in \mathtt{R}^{\mathtt{S}}$ and suppose that $\cdot \vdash e : \tau$ and $\cdot \vdash e' : \tau$. First, assume that $e\ \mathtt{R}^{\mathtt{STE}}\ e'$, we want to show that $f\ e \simeq f'\ e'$. From the definition, we know that $e \simeq e'$. Suppose neither e nor e' terminate. Then neither $f\ e$ nor $f'\ e'$ terminate, because there exist no values that e and e' step to that allow rule *Eapp*$_3$ to be applied. Now suppose both $e\downarrow$ and $e'\downarrow$. Then there exist some v, v' such that $e \mapsto^* v$ and $e' \mapsto^* v'$. By definition, $(v, v') \in \mathtt{R}^{\mathtt{ST}}$, and thus $f\ v \simeq f'\ v'$. Therefore, by rule *Hstep*, $f\ e \simeq f'\ e'$, which we have shown to be true in both cases and thus the desired result is obtained. □

Lemma 7.7. Given relations $\mathtt{R} \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_1')$ and $\mathtt{Q} \subseteq \mathtt{Val}(\tau_2) \times \mathtt{Val}(\tau_2')$ such that $\mathtt{Q} = \mathtt{Q}^{\mathtt{ST}}$, and that $(v, v') \in \mathtt{R} \Rightarrow (g\ v, g'\ v') \in \mathtt{Q}^{\mathtt{E}}$, then $(v, v') \in \mathtt{R}^{\mathtt{ST}} \Rightarrow (g\ v, g'\ v') \in \mathtt{Q}^{\mathtt{STE}} = \mathtt{Q}^{\mathtt{E}}$, where $g : \tau_1 \to \tau_2$ and $g' : \tau_1' \to \tau_2'$.

Proof. Assume that $(v, v') \in \mathtt{R}^{\mathtt{ST}}$. We want to show that $(g\ v, g'\ v') \in \mathtt{Q}^{\mathtt{E}}$. Suppose $(f, f') \in \mathtt{Q}^{\mathtt{S}}$. Now we just need to show that $f\ (g\ v) \simeq f'\ (g'\ v')$. We claim that $(f \circ g, f' \circ g') \in \mathtt{R}^{\mathtt{S}}$, which would imply this fact by the assumption that $(v, v') \in \mathtt{R}^{\mathtt{ST}}$.

Suppose $(u, u') \in \mathtt{R}$. We need to show that $f\ (g\ u) \simeq f'\ (g'\ u')$. By our assumption that $(u, u') \in \mathtt{R} \Rightarrow (g\ u, g'\ u') \in \mathtt{Q}^{\mathtt{E}}$, we know that $(g\ u, g'\ u') \in \mathtt{Q}^{\mathtt{E}}$. The result follows from applying Lemma 7.6 with $(f, f') \in \mathtt{Q}^{\mathtt{S}}$ and $(g\ u, g'\ u') \in \mathtt{Q}^{\mathtt{STE}}$. □

7.2 Admissibility

Definition Let $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$, and define the recursive functions $\cdot \vdash f : \tau''$ and $\cdot \vdash f' : \tau''$, where either $\tau'' = \tau_1 \to \tau_2$ or $\tau'' = \tau_1 \Rightarrow \tau_2$.

Given that $w : \tau'' \vdash e : \tau$ and $w : \tau'' \vdash e' : \tau'$, we say that $\mathtt{R}$ is admissible if

$$(e^{f[i]}, e'^{f'[i]}) \in \mathtt{R}^{\mathtt{E}} \text{ for all } i = 0, 1, ... \Rightarrow (e^{f[w]}, e'^{f'[w]}) \in \mathtt{R}^{\mathtt{E}}$$

Lemma 7.8. Let $\mathtt{R} \subseteq \mathtt{Val}(\tau) \times \mathtt{Val}(\tau')$ such that $\mathtt{R} = \mathtt{R}^{\mathtt{ST}}$. Then $\mathtt{R}$ is admissible.

Proof. To show the result, we need to show that $e^{f[w]} \simeq e'^{f'[w]}$ and that for all v and v', if $e^{f[w]} \mapsto^* v$ and $e'^{f'[w]} \mapsto^* v'$, then $(v, v') \in \mathtt{R}$.

By Theorem 6.6 on $e^{f[w]}$ we get that $e^{f[w]} \simeq e^{f[i]}$ for some i. By our assumption that $(e^{f[i]}, e'^{f'[i]}) \in \mathtt{R}^{\mathtt{E}}$, we get that $e^{f[i]} \simeq e'^{f'[i]}$. Then by applying Theorem 6.6 again but this time for the reverse, we get that $e'^{f'[i]} \simeq e'^{f'[w]}$. Thus by transitivity of Kleene equivalence, we have that $e^{f[w]} \simeq e'^{f'[w]}$.

Now assume $e^{f[w]} \mapsto^* v$ and $e'^{f[w]} \mapsto^* v'$ for some v, v'. We want to show that $(v, v') \in \mathtt{R}$. Let $(g, g') \in \mathtt{R}^{\mathtt{S}}$. Now if we consider $(g\ e)^{f[w]}$, by Theorem 6.6 we know that $(g\ e)^{f[w]} \simeq (g\ e)^{f[i]}$ for some i. By our assumption, we know that $(e^{f[i]}, e'^{f'[i]}) \in \mathtt{R}^{\mathtt{E}}$. Then we can apply Lemma 7.6 using the fact that $\mathtt{R}^{\mathtt{E}} = \mathtt{R}^{\mathtt{STE}}$ to get that $g\ (e^{f[i]}) \simeq g'\ (e'^{f'[i]})$. Similar to before, we can apply Theorem 6.6 to get that $(g'\ e')^{f'[i]} \simeq (g'\ e')^{f'[w]}$.

By the transitivity of Kleene equivalence, we get that $(g\ e)^{f[w]} \simeq (g'\ e')^{f'[w]}$. Since g and g' are closed, this is equivalent to $g\ e^{f[w]} \simeq g'\ e'^{f'[w]}$. We know that $e^{f[w]} \mapsto^* v$ and $e'^{f'[w]} \mapsto^* v'$, so that implies that $g\ e^{f[w]} \mapsto^* g\ v$ and $g'\ e'^{f'[w]} \mapsto^* g'\ v'$, and therefore $g\ v \simeq g'\ v'$ since they have the same termination (they are just further along in the evaluation). But since by assumption $(g, g') \in \mathtt{R}^{\mathtt{S}}$, we know that $(v, v') \in \mathtt{R}^{\mathtt{ST}} = \mathtt{R}$.

Therefore, because v and v' were arbitrary, we have that $(e^{f[w]}, e'^{f'[w]}) \in \mathtt{R}^{\mathtt{E}}$, as desired. □

8 Logical Equivalence

Definition The judgement $\delta : \Delta$ states that δ is a type substitution that assigns a closed type to each type variable $\alpha \in \Delta$. A type substitution δ induces a substitution function $\widehat{\delta}$ on types $\Delta \vdash \tau$ given by the equation

$$\widehat{\delta}(\tau) = [\delta(\alpha_1)/\alpha_1]...[\delta(\alpha_n)/\alpha_n]\tau$$

and similarly for terms. Substitution is extended to contexts pointwise by defining $\widehat{\delta}(\Gamma)(x) = \widehat{\delta}(\Gamma(x))$ for each $x \in dom(\Gamma)$.

Definition Given two type substitutions, $\delta_1 : \Delta$ and $\delta_2 : \Delta$, we define an ST-closed relation assignment, η, between δ_1 and δ_2 as an assignment of an ST-closed relation $\eta(\alpha) \in \mathtt{Val}(\delta_1(\alpha)) \times \mathtt{Val}(\delta_2(\alpha))$ to each $\alpha \in \Delta$. The judgement $\eta : \delta_1 \leftrightarrow \delta_2$ states that η is an ST-closed relation assignment between δ_1 and δ_2.

Definition Suppose $\Delta \vdash \tau\,\mathtt{type}$ for the below types. Also suppose that $\delta_1 : \Delta$ and $\delta_2 : \Delta$, and that $\eta : \delta_1 \leftrightarrow \delta_2$. We define a relation $[\![\tau]\!]$ based on types as follows

$$
\begin{array}{lcl}
[\![\alpha]\!]_{\eta:\delta_1\leftrightarrow\delta_2} & = & \eta(\alpha) \\
[\![\mathtt{unit}]\!]_{\eta:\delta_1\leftrightarrow\delta_2} & = & \{(\mathtt{()},\mathtt{()})\} \\
[\![\mathtt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2} & = & \{(v,v) \mid v \in \mathtt{Val}(\mathtt{int})\} \\
[\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2} & = & \{(v_1,v_2) \in \mathtt{Val}(\delta_1(\tau_1 \to \tau_2)) \times \mathtt{Val}(\delta_2(\tau_1 \to \tau_2)) \mid \\
& & \quad \text{if } (v_1',v_2') \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2} \text{ then } (v_1\ v_1', v_2\ v_2') \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\} \\
[\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2} & = & \{(v_1,v_2) \in \mathtt{Val}(\delta_1(\tau_1 \Rightarrow \tau_2)) \times \mathtt{Val}(\delta_2(\tau_1 \Rightarrow \tau_2)) \mid \\
& & \quad \text{if } (v_1',v_2') \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2} \text{ then } (v_1 \frown v_1', v_2 \frown v_2') \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\} \\
[\![\tau_1 \times \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2} & = & \{(v_1,v_2) \in \mathtt{Val}(\delta_1(\tau_1 \times \tau_2)) \times \mathtt{Val}(\delta_2(\tau_1 \times \tau_2)) \mid \\
& & \quad (\pi_1 v_1, \pi_1 v_2) \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2} \text{ and } (\pi_2 v_1, \pi_2 v_2) \in [\![\tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}\} \\
[\![\forall\alpha.\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2} & = & \{(v_1,v_2) \in \mathtt{Val}(\delta_1(\forall\alpha.\tau)) \times \mathtt{Val}(\delta_2(\forall\alpha.\tau)) \mid \\
& & \quad \forall \tau_1\,\mathtt{type}, \tau_2\,\mathtt{type}, \mathtt{R} \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2) \text{ for } \mathtt{R} = \mathtt{R}^{\mathtt{ST}} \\
& & \quad (v_1[\tau_1], v_2[\tau_2]) \in [\![\tau]\!]^{\mathtt{E}}_{(\eta\otimes\alpha\hookrightarrow\mathtt{R}):(\delta_1\otimes\alpha\hookrightarrow\tau_1)\leftrightarrow(\delta_2\otimes\alpha\hookrightarrow\tau_2)}\} \\
[\![\exists\alpha.\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2} & = & \{(v_1,v_2) \in \mathtt{Val}(\delta_1(\exists\alpha.\tau)) \times \mathtt{Val}(\delta_2(\exists\alpha.\tau)) \mid \\
& & \quad \exists \Delta \vdash \tau_1\,\mathtt{type}, \Delta \vdash \tau_2\,\mathtt{type}, \mathtt{R} \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2) \text{ for } \mathtt{R} = \mathtt{R}^{\mathtt{ST}} \\
& & \quad (v_1', v_2') \in [\![\tau]\!]_{(\eta\otimes\alpha\hookrightarrow\mathtt{R}):(\delta_1\otimes\alpha\hookrightarrow\tau_1)\leftrightarrow(\delta_2\otimes\alpha\hookrightarrow\tau_2)} \\
& & \quad \text{with } v_1 = \mathtt{pack}[\tau_1, v_1']\,\mathtt{as}\,\exists\alpha.\tau \text{ and } v_2 = \mathtt{pack}[\tau_2, v_2']\,\mathtt{as}\,\exists\alpha.\tau\}^{\mathtt{ST}}
\end{array}
$$

As a shorthand, when η is empty, we just write $[\![\tau]\!]$.

Lemma 8.1. For all τ, $[\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2} = [\![\tau]\!]^{\mathtt{ST}}_{\eta:\delta_1\leftrightarrow\delta_2}$.

Proof. We already know that $[\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2} \subseteq [\![\tau]\!]^{\mathtt{ST}}_{\eta:\delta_1\leftrightarrow\delta_2}$ by Lemma 7.1.

Thus it just remains to be shown that $[\![\tau]\!]^{\mathtt{ST}}_{\eta:\delta_1\leftrightarrow\delta_2} \subseteq [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, which will proceed by induction on the structure of τ.

Case for $\tau = \alpha$
By definition, we know that $[\![\alpha]\!]_{\eta:\delta_1\leftrightarrow\delta_2} = \eta(\alpha)$. The result follows from the fact that $\eta(\alpha)$ is ST-closed.

Case for $\tau = \texttt{unit}$
Trivial, because $[\![\texttt{unit}]\!]^{\texttt{ST}}_{\eta:\delta_1\leftrightarrow\delta_2}$ only relates values of type unit, and since there is only one value of type $\texttt{unit}$, $[\![\texttt{unit}]\!]^{\texttt{ST}}_{\eta:\delta_1\leftrightarrow\delta_2} = \{((),())\} = [\![\texttt{unit}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$.

Case for $\tau = \texttt{int}$
Suppose that $(v_1, v_2) \in [\![\texttt{int}]\!]^{\texttt{ST}}_{\eta:\delta_1\leftrightarrow\delta_2}$. We want to show that $(v_1, v_2) \in [\![\texttt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. First, define $f = \lambda x : \texttt{int}.\texttt{ifz}(x - v, 0, \bot)$. It must be that $(f, f) \in [\![\texttt{int}]\!]^{\texttt{S}}_{\eta:\delta_1\leftrightarrow\delta_2}$, because by the definition of $[\![\texttt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, all pairs in it are of the form $(v, v) \in [\![\texttt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, and clearly $f\ v \simeq f\ v$. Since $(f, f) \in [\![\texttt{int}]\!]^{\texttt{S}}_{\eta:\delta_1\leftrightarrow\delta_2}$, we know that $f\ v_1 \simeq f\ v_2$ by the definition of $[\![\texttt{int}]\!]^{\texttt{ST}}_{\eta:\delta_1\leftrightarrow\delta_2}$. But the only way for $f\ v_2$ to have the same halting behavior as $f\ v_1$ (which converges to 0) is if $v_1 = v_2$, as otherwise $f\ v_2$ will diverge. Thus $v_1 = v_2$, and so $(v_1, v_2) = (v_1, v_2) \in [\![\texttt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ since $v_1 \in \texttt{Val}(\texttt{int})$.

Case for $\tau = \tau_1 \to \tau_2$
We will prove this case by making use of Lemma 7.7. First we will define $\texttt{R} = [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and $\texttt{Q} = [\![\tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. By induction on τ_2 we know that $\texttt{Q} = \texttt{Q}^{\texttt{ST}}$. Also, define $g_1 = \lambda h_1 : \tau_1 \to \tau_2.h_1\ v_1$ and $g_2 = \lambda h_2 : \tau_1 \to \tau_2.h_2\ v_2$, where $(v_1, v_2) \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$.

We want to show that $(f_1, f_2) \in \texttt{R} \Rightarrow (g_1\ f_1, g_2\ f_2) \in \texttt{Q}^{\texttt{E}}$. Assume $(f_1, f_2) \in \texttt{R}$. Then $g_1\ f_1 \mapsto f_1\ v_1$ and $g_2\ f_2 \mapsto f_2\ v_2$. Since $(v_1, v_2) \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ by assumption, we know that $(f_1\ v_1, f_2\ v_2) \in \texttt{Q}^{\texttt{E}}$ by the definition of $\texttt{R} = [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. Therefore, by Corollary 7.5, $(g_1\ f_1, g_2\ f_2) \in \texttt{Q}^{\texttt{E}}$, as desired.

Then by Lemma 7.7, we know that $(f_1, f_2) \in \texttt{R}^{\texttt{ST}} \Rightarrow (g_1\ f_1, g_2\ f_2) \in \texttt{Q}^{\texttt{E}}$. Assume that $(f_1, f_2) \in \texttt{R}^{\texttt{ST}}$. To show this case, we need to show that $(f_1, f_2) \in \texttt{R}$. However, by the above we know that $(g_1\ f_1, g_2\ f_2) \in \texttt{Q}^{\texttt{E}}$, and thus $(f_1\ v_1, f_2\ v_2) \in \texttt{Q}^{\texttt{E}}$ by Corollary 7.5. Then by the definition of $[\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, $(f_1, f_2) \in \texttt{R}$, and so the result follows.

Case for $\tau = \tau_1 \Rightarrow \tau_2$
Essentially the same as the case for $\tau = \tau_1 \to \tau_2$.

Case for $\tau = \tau_1 \times \tau_2$
We will prove this case similarly to the $\tau_1 \to \tau_2$ case in that we'll make use of Lemma 7.7, however it will set it up slightly differently. Define $\texttt{R} = [\![\tau_1 \times \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and $\texttt{Q}_1 = [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. By induction we know that $\texttt{Q}_1 = \texttt{Q}_1^{\texttt{ST}}$. Also, define $g_1 = \lambda h_1 : \tau_1 \times \tau_2.\pi_1 h_1$ and $g_2 = \lambda h_2 : \tau_1 \times \tau_2.\pi_1 h_2$.

We want to show that $(v_1, v_2) \in \texttt{R} \Rightarrow (g_1\ v_1, g_2\ v_2) \in \texttt{Q}^{\texttt{E}}$. Assume $(v_1, v_2) \in \texttt{R}$. We know that $g_1\ v_1 \mapsto \pi_1 v_1$ and $g_2\ v_2 \mapsto \pi_1 v_2$. By the definition of $\texttt{R} = [\![\tau_1 \times \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, we know that $(\pi_1 v_1, \pi_1 v_2) \in \texttt{Q}_1^{\texttt{E}}$. Therefore, by Corollary 7.5, we get that $(g_1\ v_1, g_2\ v_2) \in \texttt{Q}_1^{\texttt{E}}$.

Using this fact, we can apply Lemma 7.7 to get that $(v_1, v_2) \in \texttt{R}^{\texttt{ST}} \Rightarrow (g_1\ v_1, g_2\ v_2) \in \texttt{Q}_1^{\texttt{E}}$. Thus, if we assume that $(v_1, v_2) \in \texttt{R}^{\texttt{ST}}$, we want to show that $(v_1, v_2) \in \texttt{R}$. By the above, we get that $(g_1\ v_1, g_2\ v_2) \in \texttt{Q}_1^{\texttt{E}}$. Again using Corollary 7.5, we get that $(\pi_1 v_1, \pi_1 v_2) \in \texttt{Q}_1^{\texttt{E}}$.

We can then do the same as above but this time using $\texttt{Q}_2 = [\![\tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and show that $(\pi_2 v_1, \pi_2 v_2) \in \texttt{Q}_2^{\texttt{E}}$. This combined with the above fact shows us that by definition $(v_1, v_2) \in \texttt{R}$, as desired.

Case for $\tau = \forall\alpha.\tau'$
Again we will make use of Lemma 7.7. Define $\texttt{R} = [\![\forall\alpha.\tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. Also, for $\cdot \vdash \tau_1\ \texttt{type}$, $\cdot \vdash \tau_2\ \texttt{type}$

and $\mathtt{R}' \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$ such that $\mathtt{R}' = \mathtt{R}'^{\mathtt{ST}}$ define $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}'$, $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \tau_1$ and $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \tau_2$. Then let $\mathtt{Q} = [\![\tau']\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$, as well as $g_1 = \lambda h_1 : \forall \alpha.\tau'.h_1[\tau_1]$ and $g_2 = \lambda h_2 : \forall \alpha.\tau'.h_2[\tau_2]$.

To make use of the lemma, we must show that $(v_1, v_2) \in \mathtt{R} \Rightarrow (g_1\ v_1, g_2\ v_2) \in \mathtt{Q}^{\mathtt{E}}$. Assume that $(v_1, v_2) \in \mathtt{R}$. We know that $g_1\ v_1 \mapsto v_1[\tau_1]$ and $g_2\ v_2 \mapsto v_2[\tau_2]$. By the definition of $\mathtt{R} = [\![\forall \alpha.\tau']\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$, we know that $(v_1[\tau_1], v_2[\tau_2]) \in \mathtt{Q}^{\mathtt{E}}$ since τ_1 and τ_2 were arbitrary. Therefore, by Corollary 7.5, we get that $(g_1\ v_1, g_2\ v_2) \in \mathtt{Q}^{\mathtt{E}}$.

Using this fact, we can apply Lemma 7.7 to get that $(v_1, v_2) \in \mathtt{R}^{\mathtt{ST}} \Rightarrow (g_1\ v_1, g_2\ v_2) \in \mathtt{Q}^{\mathtt{E}}$. Thus, if we assume that $(v_1, v_2) \in \mathtt{R}^{\mathtt{ST}}$, we want to show that $(v_1, v_2) \in \mathtt{R}$. By the above, we get that $(g_1\ v_1, g_2\ v_2) \in \mathtt{Q}^{\mathtt{E}}$. Again using Corollary 7.5, we get that $(v_1[\tau_1], v_2[\tau_2]) \in \mathtt{Q}^{\mathtt{E}}$. Then by definition, we have that $(v_1, v_2) \in \mathtt{R}$.

Case for $\tau = \exists \alpha.\tau'$
Unfortunately, we can't use the same strategies as in the previous cases to prove this case. Thus, we have to build the ST-closure into the definition. Since $[\![\exists \alpha.\tau']\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ is defined as the ST-closure of something, and since we previously proved that the ST-closure is idemptotent, we have that $[\![\exists \alpha.\tau']\!]^{\mathtt{ST}}_{\eta:\delta_1 \leftrightarrow \delta_2} = [\![\exists \alpha.\tau']\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$, as desired.

□

Corollary 8.2. For all τ, $[\![\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ is admissible.

Proof. Follows immediately from Lemma 8.1 and Lemma 7.8. □

Lemma 8.3. For $[\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, we can essentially use the definition as if it were $[\![\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. More specifically, the following are true:

If $(\pi_1 e_1, \pi_1 e_2) \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$ and $(\pi_2 e_1, \pi_2 e_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, then $(e_1, e_2) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$.

If $(e_1, e_2) \in [\![\tau_1 \to \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, then if $(e_1', e_2') \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, $(e_1\ e_1', e_2\ e_2') \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$.

If $(e_1, e_2) \in [\![\tau]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$, where $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \tau_1$, $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \tau_2$, $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}$ for an ST-closed $\mathtt{R} \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$, then $(\mathtt{pack}[\tau_1, e_1]\ \mathtt{as}\ \exists \alpha.\tau, \mathtt{pack}[\tau_2, e_2]\ \mathtt{as}\ \exists \alpha.\tau) \in [\![\exists \alpha.\tau]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$.

Proof. **Case for** $\tau_1 \times \tau_2$
Suppose that $(\pi_1 e_1, \pi_1 e_2) \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$ and $(\pi_2 e_1, \pi_2 e_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, we want to show that $(e_1, e_2) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$. If e_1 does not terminate, then clearly neither do $\pi_1 e_1$ and $\pi_2 e_1$. By our assumptions this implies that neither $\pi_1 e_2$ nor $\pi_2 e_2$ terminate. This implies that e_2 doesn't terminate (and the same can be shown in the other direction). Thus the result holds in this case.

Suppose that $e_1 \downarrow$, so $e_1 \mapsto^* \langle v_1, v_2 \rangle$, where $v_1\ \mathtt{val}$ and $v_2\ \mathtt{val}$. Then clearly $\pi_1 e_1 \mapsto^* v_1$ and $\pi_2 e_1 \mapsto^* v_2$. Then by our assumptions we know that $\pi_1 e_2 \mapsto^* v_1'$ and $\pi_2 e_2 \mapsto^* v_2'$, where $(v_1, v_1') \in [\![\tau_1]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ and $(v_2, v_2') \in [\![\tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. Then by definition $(e_1, e_2) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, as desired.

Case for $\tau_1 \to \tau_2$
Suppose that $(e_1, e_2) \in [\![\tau_1 \to \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$ and $(e_1', e_2') \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, and we want to show that $(e_1\ e_1', e_2\ e_2') \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$. Suppose that $e_1\ e_1' \downarrow$. This implies that $e_1 \downarrow$ and $e_1' \downarrow$, which by the above implies that $e_2 \downarrow$ and $e_2' \downarrow$. Thus we have that $e_1 \mapsto^* v_1$, $e_1' \mapsto^* v_1'$, $e_2 \mapsto^* v_2$, and $e_2' \mapsto^* v_2'$ for values v_1, v_1', v_2, v_2'. This tells us that $(v_1, v_2) \in [\![\tau_1 \to \tau_2]\!]$ and $(v_1', v_2') \in [\![\tau_1]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$, so by definition $(v_1\ v_1', v_2\ v_2') \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$. However, we know that $e_1\ e_1' \mapsto^* v_1\ v_1'$ and $e_2\ e_2' \mapsto^* v_2\ v_2'$, which

implies that $(e_1\ e_1', e_2\ e_2') \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$, as desired. This also tells us that $e_2\ e_2' \downarrow$. Similarly we can show this assuming that $e_2\ e_2' \downarrow$, which will tell us that $e_1\ e_1' \downarrow$. Thus we have that $e_1\ e_1' \simeq e_2\ e_2'$, and then if they both terminate the above gets us the desired result.

Case for $\exists\alpha.\tau$
Suppose $(e_1, e_2) \in [\![\tau]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$, where $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \tau_1$, $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \tau_2$, $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}$ for an ST-closed $\mathtt{R} \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$, and we want to show that

$$(\mathtt{pack}[\tau_1, e_1]\,\mathtt{as}\,\exists\alpha.\tau, \mathtt{pack}[\tau_2, e_2]\,\mathtt{as}\,\exists\alpha.\tau) \in [\![\exists\alpha.\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

First, if $\mathtt{pack}[\tau_1, e_1]\,\mathtt{as}\,\exists\alpha.\tau$ doesn't terminate, then clearly e_1 doesn't terminate. By assumption, this implies that e_2 doesn't terminate, which then implies that $\mathtt{pack}[\tau_2, e_2]\,\mathtt{as}\,\exists\alpha.\tau$ doesn't as well.

If $\mathtt{pack}[\tau_1, e_1]\,\mathtt{as}\,\exists\alpha.\tau$ does terminate, then we know that

$$\mathtt{pack}[\tau_1, e_1]\,\mathtt{as}\,\exists\alpha.\tau \mapsto^* \mathtt{pack}[\tau_1, v_1]\,\mathtt{as}\,\exists\alpha.\tau$$

for some v_1. But this means that $e_1 \mapsto^* v_1$, which means that by assumption there must be some v_2 such that $e_2 \mapsto^* v_2$, and $(v_1, v_2) \in [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. But then by definition of the type relation,

$$(\mathtt{pack}[\tau_1, v_1]\,\mathtt{as}\,\exists\alpha.\tau, \mathtt{pack}[\tau_2, v_2]\,\mathtt{as}\,\exists\alpha.\tau) \in [\![\exists\alpha.\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$$

So the desired result follows.

□

Definition Recalling our definition of expression substitutions γ from earlier, we define the relation $\gamma \sim_\Gamma \gamma'[\eta : \delta_1 \leftrightarrow \delta_2]$ to mean that $\gamma : \Gamma$, $\gamma' : \Gamma$, and that for all $x \in \Gamma$, $(\gamma(x), \gamma'(x)) \in [\![\Gamma(x)]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$.

Definition We say that the expressions $\Delta; \Gamma \vdash e_1 : \tau$ and $\Delta; \Gamma \vdash e_2 : \tau$ are logically equivalent, written $\Delta; \Gamma \vdash e_1 \sim e_2 : \tau$ iff, for every assignment $\delta_1 : \Delta$ and $\delta_2 : \Delta$, and every ST-closed relation assignment $\eta : \delta_1 \leftrightarrow \delta_2$, if $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$ then $(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$.

As a shorthand, we write $e_1 \sim_\tau e_2$ instead of $\cdot \vdash e_1 \sim e_2 : \tau$.

8.1 Compositionality

Lemma 8.4. If $[\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2} = [\![\tau']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, then $[\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2} = [\![\tau']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$,

Proof. Suppose that $(e_1, e_2) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. If e_1 does not halt, then we know that e_2 also does not halt, and thus $(e_1, e_2) \in [\![\tau']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$. However, if $e_1 \downarrow$, then also $e_2 \downarrow$, so we know that $e_1 \mapsto^* v_1$ and $e_2 \mapsto^* v_2$, and thus $(v_1, v_2) \in [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. But by assumption, $(v_1, v_2) \in [\![\tau']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and thus by definition $(e_1, e_2) \in [\![\tau']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$. □

Lemma 8.5. Suppose $\Delta \vdash \tau'$ and $\delta_1 : \Delta$, $\delta_2 : \Delta$, and $\eta : \delta_1 \leftrightarrow \delta_2$. Let $\mathtt{R} = [\![\tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. Then $(v_1, v_2) \in [\![[\tau'/\alpha]\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ if and only if $(v_1, v_2) \in [\![\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, where $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}$, $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \delta_1(\tau')$, and $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \delta_2(\tau')$.

Proof. By induction on the structure of τ.

Case for $\tau = \alpha$
Since $\tau = \alpha$, $[\![\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'} = \mathtt{R}$, where as defined above $\mathtt{R} = [\![\tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2} = [\![[\tau'/\alpha]\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, so these are the same and the desired result holds.

Case for $\tau = \alpha' \neq \alpha$
Since $\alpha' \neq \alpha$, we know that $\alpha' \in \eta$, so we have that both $[\![\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'} = \eta(\alpha')$ and $[\![[\tau'/\alpha]\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'} = \eta(\alpha')$, so these are the same and the desired result holds.

Case for $\tau = \mathtt{unit}$
Trivially true, since the logical relation for `unit` does not depend on type variables.

Case for $\tau = \mathtt{int}$
Trivially true, since the logical relation for `int` does not depend on type variables.

Case for $\tau = \tau_1 \to \tau_2$
Assume $(v_1, v_2) \in [\![[\tau'/\alpha]\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, which tells us that

$$(v_1', v_2') \in [\![[\tau'/\alpha]\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2} \Rightarrow (v_1\ v_1', v_2\ v_2') \in [\![[\tau'/\alpha]\tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$$

Assuming that $(v_1', v_2') \in [\![\tau_1]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, we know by induction that $(v_1', v_2') \in [\![[\tau'/\alpha]\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and then by the above that $(v_1\ v_1', v_2\ v_2') \in [\![[\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. Then by induction and Lemma 8.4 we know that $(v_1\ v_1', v_2\ v_2') \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$. By the definition of the logical relation this means that $(v_1, v_2) \in [\![\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, as desired.

The opposite direction is shown in a similar manner.

Case for $\tau = \tau_1 \Rightarrow \tau_2$
Assume $(v_1, v_2) \in [\![[\tau'/\alpha]\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, which tells us that

$$(v_1', v_2') \in [\![[\tau'/\alpha]\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2} \Rightarrow (v_1 \frown v_1', v_2 \frown v_2') \in [\![[\tau'/\alpha]\tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$$

Assuming that $(v_1', v_2') \in [\![\tau_1]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, we know by induction that $(v_1', v_2') \in [\![[\tau'/\alpha]\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and then by the above that $(v_1 \frown v_1', v_2 \frown v_2') \in [\![[\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. Then by induction and Lemma 8.4 we know that $(v_1 \frown v_1', v_2 \frown v_2') \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$. By the definition of the logical relation this means that $(v_1, v_2) \in [\![\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, as desired.

The opposite direction is shown in a similar manner.

Case for $\tau = \tau_1 \times \tau_2$
Assume $(v_1, v_2) \in [\![[\tau'/\alpha]\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, which tells us that $(\pi_1 v_1, \pi_1 v_2) \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and $(\pi_2 v_1, \pi_2 v_2) \in [\![\tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. However, by induction we get that $(\pi_1 v_1, \pi_1 v_2) \in [\![[\tau'/\alpha]\tau_1]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$ and $(\pi_2 v_1, \pi_2 v_2) \in [\![[\tau'/\alpha]\tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$. But this means that $(v_1, v_2) \in [\![[\tau'/\alpha]\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, as desired.

The opposite direction is shown in a similar manner.

Case for $\tau = \forall\beta.\tau''$
Assume $(v_1, v_2) \in [\![[\tau'/\alpha]\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, which tells us that $\forall \tau_1\ \mathtt{type}, \tau_2\ \mathtt{type}, \mathtt{R} \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$ such that $\mathtt{R}' = \mathtt{R}'^{\mathtt{ST}}$,

$$(v_1[\tau_1], v_2[\tau_2]) \in [\![[\tau'/\alpha]\tau'']\!]^{\mathtt{E}}_{\eta\otimes\beta\hookrightarrow\mathtt{R}':\delta_1\otimes\beta\hookrightarrow\tau_1\leftrightarrow\delta_2\otimes\beta\hookrightarrow\tau_2}$$

But by induction and Lemma 8.4, we know that

$$(v_1[\tau_1], v_2[\tau_2]) \in [\![\tau'']\!]^{\mathtt{E}}_{\eta' \otimes \beta \hookrightarrow \mathtt{R}' : \delta_1' \otimes \beta \hookrightarrow \tau_1 \leftrightarrow \delta_2' \otimes \beta \hookrightarrow \tau_2}$$

for the above, which then tells us that $(v_1, v_2) \in [\![\tau]\!]_{\eta' : \delta_1' \leftrightarrow \delta_2'}$ by definition.

Case for $\tau = \exists \beta . \tau''$
First, we will prove this true for $\mathtt{Q}$, where $\mathtt{Q}$ is equivalent to $[\![[\tau'/\alpha]\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ except without the additional ST-closure wrapped around it. Similarly define $\mathtt{Q}'$ in relation to $[\![\tau]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$. Assume $(v_1, v_2) \in \mathtt{Q}$, which tells us that there exist $\tau_1\,\mathtt{type}, \tau_2\,\mathtt{type}$, $\mathtt{R} \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$ such that $\mathtt{R}' = \mathtt{R}'^{\mathtt{ST}}$, where

$$(v_1', v_2') \in [\![[\tau'/\alpha]\tau'']\!]_{\eta \otimes \beta \hookrightarrow \mathtt{R}' : \delta_1 \otimes \beta \hookrightarrow \tau_1 \leftrightarrow \delta_2 \otimes \beta \hookrightarrow \tau_2}$$

with $v_1 = \mathtt{pack}[\tau_1, v_1']\,\mathtt{as}\,\exists \beta.\tau''$ and $v_2 = \mathtt{pack}[\tau_2, v_2']\,\mathtt{as}\,\exists\beta.\tau''$. But then by induction we get that

$$(v_1', v_2') \in [\![\tau'']\!]_{\eta' \otimes \beta \hookrightarrow \mathtt{R}' : \delta_1' \otimes \beta \hookrightarrow \tau_1 \leftrightarrow \delta_2' \otimes \beta \hookrightarrow \tau_2}$$

Then by definition, $(v_1, v_2) \in \mathtt{Q}'$. We can show the reverse direction in a similar manner.

We have shown that $\mathtt{Q} = \mathtt{Q}'$, so then by Lemma 8.4, we know that $\mathtt{Q}^{\mathtt{ST}} = \mathtt{Q}'^{\mathtt{ST}}$, which is equivalent to saying that $[\![[\tau'/\alpha]\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2} = [\![\tau]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$, which is what we wanted to show.

□

Lemma 8.6. Suppose $\Delta \vdash \tau'$ and $\delta_1 : \Delta$, $\delta_2 : \Delta$, and $\eta : \delta_1 \leftrightarrow \delta_2$. Let $\mathtt{R} = [\![\tau']\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. Then $(v_1, v_2) \in [\![[\tau'/\alpha]\tau]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$ if and only if $(v_1, v_2) \in [\![\tau]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$, where $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}$, $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \delta_1(\tau')$, and $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \delta_2(\tau')$.

Proof. Follows immediately from Lemma 8.5 and Lemma 8.4. □

9 Logical and Contextual Equivalence Coincide

9.1 Reflexivity

Theorem 9.1. If $\Delta; \Gamma \vdash e : \tau$, then $\Delta; \Gamma \vdash e \sim e : \tau$.

Proof. By induction over typing rules. For each case we simply need to show that given substitutions δ_1 and δ_2 and an ST-closed relation assignment $\eta : \delta_1 \leftrightarrow \delta_2$ such that $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$, then $(\widehat{\gamma}_1(\delta_1(e_1)), \widehat{\gamma}_2(\delta_2(e_2))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$.

Case for $Tunit$
Trivially true by definition of logical equivalence, as clearly $(\mathtt{()}, \mathtt{()}) \in [\![\mathtt{unit}]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$.

Case for $Tvar$
Assume that $\Delta; \Gamma \vdash x : \tau$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. Since x is a variable, clearly $\widehat{\delta}_1(x) = x$ and $\widehat{\delta}_2(x) = x$. Thus, since $x \in \Gamma$, for some e, e' we know that $\gamma_1(x) = e_1$ and $\gamma_2(x) = e_2$. By the definition of $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$ we know that $(e_1, e_2) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$, which by the above is equivalent to $(\widehat{\gamma}_1(\widehat{\delta}_1(x)), \widehat{\gamma}_2(\widehat{\delta}_2(x))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. Thus we have the desired result.

Case for $Tint$
Assume that $\Delta; \Gamma \vdash n : \mathtt{int}$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. By definition, $(n, n) \in [\![\mathtt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and thus $(n, n) \in [\![\mathtt{int}]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. Since n is a closed term and contains no term or type variables, we know that $\widehat{\gamma}_1(\widehat{\delta}_1(n)) = n$ and $\widehat{\gamma}_2(\widehat{\delta}_2(n)) = n$, so we have that $(\widehat{\gamma}_1(\widehat{\delta}_1(n)), \widehat{\gamma}_2(\widehat{\delta}_2(n))) \in [\![\mathtt{int}]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$.

Case for $Tintop$
Assume that $\Delta; \Gamma \vdash e_1 \, \mathtt{p} \, e_2 : \mathtt{int}$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. By induction we get that $\Delta; \Gamma \vdash e_1 \sim e_1 : \mathtt{int}$ and $\Delta; \Gamma \vdash e_2 \sim e_2 : \mathtt{int}$, or equivalently

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))) \in [\![\mathtt{int}]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\mathtt{int}]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

This means that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e_1))$ and $\widehat{\gamma}_1(\widehat{\delta}_1(e_2)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e_2))$. From this we can see that

$$\widehat{\gamma}_1(\widehat{\delta}_1(e_1 \, \mathtt{p} \, e_2)) \downarrow \Leftrightarrow \widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \downarrow \text{ and } \widehat{\gamma}_1(\widehat{\delta}_1(e_2)) \downarrow$$

$$\Leftrightarrow \widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \downarrow \text{ and } \widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \downarrow \Leftrightarrow \widehat{\gamma}_2(\widehat{\delta}_2(e_1 \, \mathtt{p} \, e_2)) \downarrow$$

Therefore, $\widehat{\gamma}_1(\widehat{\delta}_1(e_1 \, \mathtt{p} \, e_2)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e_1 \, \mathtt{p} \, e_2))$, which gets us the first part of the desired result.

Now, suppose that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \mapsto^* n_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \mapsto^* n_1'$ for some n_1, n_1'. By the fact that $(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))) \in [\![int]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$ by induction, we know that $(n_1, n_1') \in [\![\mathtt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ by the definition of the extension to term relations. But by the definition of $[\![\mathtt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, this means that $n_1 = n_1'$. We can show the same for e_2 with some n_2. Define $n = n_1 \, \mathtt{p} \, n_2$. Since they are substitutions, we know that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1 \, \mathtt{p} \, e_2))$ is equivalent to $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \, \mathtt{p} \, \widehat{\gamma}_1(\widehat{\delta}_1(e_2))$ and similarly that $\widehat{\gamma}_2(\widehat{\delta}_2(e_1 \, \mathtt{p} \, e_2))$ is equivalent to $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \, \mathtt{p} \, \widehat{\gamma}_2(\widehat{\delta}_2(e_2))$. Thus by the *Eintop* rules we know that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1 \, \mathtt{p} \, e_2)) \mapsto^* n$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1 \, \mathtt{p} \, e_2)) \mapsto^* n$. By definition, $(n, n) \in [\![\mathtt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, so we have shown the second half of the result, and thus

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1 \, \mathtt{p} \, e_2)), (\widehat{\gamma}_2(\widehat{\delta}_2(e_1 \, \mathtt{p} \, e_2)) \in [\![\mathtt{int}]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

Case for $Tifz$
Assume that $\Delta;\Gamma \vdash \texttt{ifz}(e_1, e_2, e_3) : \tau$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We want to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\texttt{ifz}(e_1, e_2, e_3))), \widehat{\gamma}_2(\widehat{\delta}_2(\texttt{ifz}(e_1, e_2, e_3))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

or equivalently

$$(\texttt{ifz}(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_1(\widehat{\delta}_1(e_3))))), \texttt{ifz}(\widehat{\gamma}_2(\widehat{\delta}_2(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_3))))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

By induction we get that $\Delta;\Gamma \vdash e_1 \sim e_1 : \texttt{int}$, $\Delta;\Gamma \vdash e_2 \sim e_2 : \tau$, and $\Delta;\Gamma \vdash e_3 \sim e_3 : \tau$, so this tells us that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))) \in [\![\texttt{int}]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$
$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$
$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_3)), \widehat{\gamma}_2(\widehat{\delta}_2(e_3))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

From this we get that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e_1))$. Then if $\widehat{\gamma}_1(\widehat{\delta}_1(e_1))$ doesn't terminate, then neither does $\widehat{\gamma}_2(\widehat{\delta}_2(e_1))$, and by rule $Eifz_1$ neither does $\widehat{\gamma}_1(\widehat{\delta}_1(\texttt{ifz}(e_1, e_2, e_3)))$ nor $\widehat{\gamma}_2(\widehat{\delta}_2(\texttt{ifz}(e_1, e_2, e_3)))$, so the result holds in that case. If $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \downarrow$, then also $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \downarrow$, so there exist some n, n' such that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \mapsto^* n$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \mapsto^* n'$ such that $(n, n') \in [\![\texttt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. However, this implies that $n = n'$ by the definition of $[\![\texttt{int}]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$.

If $n = 0$, then by rule $Eifz_2$ we know that both

$$\texttt{ifz}(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_1(\widehat{\delta}_1(e_3))))) \mapsto^* \widehat{\gamma}_1(\widehat{\delta}_1(e_2))$$
$$\texttt{ifz}(\widehat{\gamma}_2(\widehat{\delta}_2(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_3))))) \mapsto^* \widehat{\gamma}_2(\widehat{\delta}_2(e_2))$$

Then since we know by induction that $(\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$, it clearly follows by Corollary 7.5 that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\texttt{ifz}(e_1, e_2, e_3))), \widehat{\gamma}_2(\widehat{\delta}_2(\texttt{ifz}(e_1, e_2, e_3))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

We can do similarly in the case where $n \neq 0$ except with e_3 instead of e_2, which gets us the desired result.

Case for $Tfun$
Assume that $\Delta;\Gamma \vdash \texttt{fun}\ g(x : \tau_1).e : \tau_1 \to \tau_2$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We need to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\texttt{fun}\ g(x : \tau_1).e)), \widehat{\gamma}_2(\widehat{\delta}_2(\texttt{fun}\ g(x : \tau_1).e))) \in [\![\tau_1 \to \tau_2]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

which is equivalent to showing

$$(\texttt{fun}\ g(x : \tau_1).\widehat{\gamma}_1(\widehat{\delta}_1(e)), \texttt{fun}\ g(x : \tau_1).\widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\tau_1 \to \tau_2]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

Since this is a value already, we can just show that it is in $[\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ as opposed to $[\![\tau_1 \to \tau_2]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$, since the former implies the latter.

Define $f_1 = \texttt{fun}\ g(x : \tau_1).\widehat{\gamma}_1(\widehat{\delta}_1(e))$ and $f_2 = \texttt{fun}\ g(x : \tau_1).\widehat{\gamma}_2(\widehat{\delta}_2(e))$. We will prove by induction that for all i, $(f_1^i, f_2^i) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. In the following, assume that $(v_1, v_2) \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$.

When $i = 0$, we want to show that $(f_1^0\ v_1, f_2^0\ v_2) \in [\![\tau_2]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$ from which the result follows by definition. By definition, neither terminate, thus we trivially have that $(f_1^0\ v_1, f_2^0\ v_2) \in [\![\tau_2]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$ since $f_1^0\ v_1 \simeq f_2^0\ v_2$ and neither steps to a value.

Now suppose that we know that $(f_1^i, f_2^i) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. We want to show that $(f_1^{i+1}\ v_1, f_2^{i+1}\ v_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. By using Corollary 7.5, we just have to show that

$$([f_1^{i+1}/g][v_1/x][f_1^i/f]\widehat{\gamma}_1(\widehat{\delta}_1(e)), [f_2^{i+1}/g][v_2/x][f_2^i/f]\widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

Since the inner substitution for f was done in the definition of f_1^{i+1} and f_2^{i+1}, the outer substitution has nothing to substitute for and so we can remove it like so:

$$([v_1/x][f_1^i/g]\widehat{\gamma}_1(\widehat{\delta}_1(e)), [v_2/x][f_2^i/g]\widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

Now define $\gamma_1' = \gamma_1 \otimes g \hookrightarrow f_1^i \otimes x \hookrightarrow v_1$ and $\gamma_2' = \gamma_2 \otimes g \hookrightarrow f_2^i \otimes x \hookrightarrow v_2$. By the inner induction on n, we know that $(f_1^i, f_2^i) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and so $(f_1^i, f_2^i) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. Similarly, we also know that $(v_1, v_2) \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. Thus, for $\Gamma' = \Gamma, x : \tau_1, g : \tau_1 \to \tau_2$, we know that $\gamma_1' \sim_{\Gamma'} \gamma_2'[\eta : \delta_1 \leftrightarrow \delta_2]$. But by our outer induction, we get that $\Delta; \Gamma \vdash e \sim e : \tau_2$, which tells us that $(\widehat{\gamma}_1'(\widehat{\delta}_1(e)), \widehat{\gamma}_2'(\widehat{\delta}_2'(e))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$, which is exactly what we wanted to show.

Now that we know that $(f_1^i, f_2^i) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ for all i, this is equivalent to saying that $(w^{f_1[i]}, w^{f_2[i]}) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ for all i. We know by Corollary 8.2 that $[\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ is admissible, so by using the property of admissibility with $e = w$ we get that $(w^{f_1[w]}, w^{f_2[w]}) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ or equivalently that $(f_1, f_2) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, which is what we wanted to show.

Case for *Tapp*
Assume that $\Delta; \Gamma \vdash e_1\ e_2 : \tau_2$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We need to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1\ e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1\ e_2))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

which is equivalent to showing that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1))\ \widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))\ \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

By induction we get that $\Delta; \Gamma \vdash e_1 \sim e_1 : \tau_1 \to \tau_2$ and $\Delta; \Gamma \vdash e_2 \sim e_2 : \tau_1$, which tell us that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))) \in [\![\tau_1 \to \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

This implies that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e_1))$. Thus if $\widehat{\gamma}_1(\widehat{\delta}_1(e_1))$ does not terminate, then neither will $\widehat{\gamma}_2(\widehat{\delta}_2(e_1))$ by the above, and thus neither $\widehat{\gamma}_1(\widehat{\delta}_1(e_1\ e_2))$ nor $\widehat{\gamma}_2(\widehat{\delta}_2(e_1\ e_2))$ will by rule $Eapp_1$. The same can be said about $\widehat{\gamma}_1(\widehat{\delta}_1(e_2))$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_2))$. Thus the result holds in these cases, so we just have to show it holds when all components terminate.

Now suppose that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \mapsto^* f_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \mapsto^* f_2$ for some f_1, f_2, as well as $\widehat{\gamma}_1(\widehat{\delta}_1(e_2)) \mapsto^* v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \mapsto^* v_2$ for some v, v'. But by the definition of the extension of the logical relation to terms, we know from the above that $(f_1, f_2) \in [\![\tau_1 \to \tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and similarly that $(v_1, v_2) \in [\![\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. Then we know that $(f_1\ v_1, f_2\ v_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$ by the definition of the logical type relation over arrow types. But then $\widehat{\gamma}_1(\widehat{\delta}_1(e_1\ e_2)) \mapsto^* f_1\ v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1\ e_2)) \mapsto^* f_2\ v_2$ by our assumptions above and the *Eapp* rules. From this we get that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1\ e_2), \widehat{\gamma}_2(\widehat{\delta}_2(e_1\ e_2)) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

as desired, by application of Corollary 7.5.

Case for $Tccfun$
Assume that $\Delta; \Gamma \vdash \widehat{\texttt{fun}}\ g(x : \tau_1).e : \tau_1 \Rightarrow \tau_2$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We need to show that

$$(\widehat{\gamma_1}(\widehat{\delta_1}(\widehat{\texttt{fun}}\ g(x : \tau_1).e)), \widehat{\gamma_2}(\widehat{\delta_2}(\widehat{\texttt{fun}}\ g(x : \tau_1).e))) \in [\![\tau_1 \Rightarrow \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

which is equivalent to showing

$$(\widehat{\texttt{fun}}\ g(x : \tau_1).\widehat{\gamma_1}(\widehat{\delta_1}(e)), \widehat{\texttt{fun}}\ g(x : \tau_1).\widehat{\gamma_2}(\widehat{\delta_2}(e))) \in [\![\tau_1 \Rightarrow \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

Since this is a value already, we can just show that it is in $[\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ as opposed to $[\![\tau_1 \Rightarrow \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, since the former implies the latter.

Define $f_1 = \widehat{\texttt{fun}}\ g(x : \tau_1).\widehat{\gamma_1}(\widehat{\delta_1}(e))$ and $f_2 = \widehat{\texttt{fun}}\ g(x : \tau_1).\widehat{\gamma_2}(\widehat{\delta_2}(e))$. We will prove by induction that for all i, $(f_1^i, f_2^i) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. In the following, assume that $(v_1, v_2) \in [\![\tau_1]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$.

When $i = 0$, we want to show that $(f_1^0\,\widehat{}\,v_1, f_2^0\,\widehat{}\,v_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$ from which the result follows by definition. By definition, neither terminate, thus we trivially have that $(f_1^0\,\widehat{}\,v_1, f_2^0\,\widehat{}\,v_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$ since $f_1^0\,\widehat{}\,v_1 \simeq f_2^0\,\widehat{}\,v_2$ and neither steps to a value.

Now suppose that we know that $(f_1^i, f_2^i) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. We want to show that $(f_1^{i+1}\,\widehat{}\,v_1, f_2^{i+1}\,\widehat{}\,v_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$. By using Corollary 7.5, we just have to show that

$$([f_1^{i+1}/g][v_1/x][f_1^i/f]\widehat{\gamma_1}(\widehat{\delta_1}(e)), [f_2^{i+1}/g][v_2/x][f_2^i/f]\widehat{\gamma_2}(\widehat{\delta_2}(e))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

Since the inner substitution for f was done in the definition of f_1^{i+1} and f_2^{i+1}, the outer substitution has nothing to substitute for and so we can remove it like so:

$$([v_1/x][f_1^i/g]\widehat{\gamma_1}(\widehat{\delta_1}(e)), [v_2/x][f_2^i/g]\widehat{\gamma_2}(\widehat{\delta_2}(e))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

Now define $\gamma_1' = g \hookrightarrow f_1^i \otimes x \hookrightarrow v_1$ and $\gamma_2' = g \hookrightarrow f_2^i \otimes x \hookrightarrow v_2$. By the inner induction on n, we know that $(f_1^i, f_2^i) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ and so $(f_1^i, f_2^i) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. Similarly, we also know that $(v_1, v_2) \in [\![\tau_1]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. Thus, for $\Gamma' = g : \tau_1 \Rightarrow \tau_2, x : \tau_1$, we know that $\gamma_1' \sim_{\Gamma'} \gamma_2'[\eta : \delta_1 \leftrightarrow \delta_2]$. But by our outer induction, we get that $\Delta; \Gamma' \vdash e \sim e : \tau_2$, which tells us that $(\widehat{\gamma_1'}(\widehat{\delta_1'}(e)), \widehat{\gamma_2'}(\widehat{\delta_2'}(e))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, which is exactly what we wanted to show, since the other substitutions don't have any effect.

Now that we know that $(f_1^i, f_2^i) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ for all i, this is equivalent to saying that $(w^{f_1[i]}, w^{f_2[i]}) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ for all i. We know by Corollary 8.2 that $[\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ is admissible, so by using the property of admissibility with $e = w$ we get that $(w^{f_1[w]}, w^{f_2[w]}) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ or equivalently that $(f_1, f_2) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$, which is what we wanted to show.

Case for $Tccapp$
Assume that $\Delta; \Gamma \vdash e_1\,\widehat{}\,e_2 : \tau_2$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We need to show that

$$(\widehat{\gamma_1}(\widehat{\delta_1}(e_1\,\widehat{}\,e_2)), \widehat{\gamma_2}(\widehat{\delta_2}(e_1\,\widehat{}\,e_2))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

which is equivalent to showing that

$$(\widehat{\gamma_1}(\widehat{\delta_1}(e_1))\,\widehat{}\,\widehat{\gamma_1}(\widehat{\delta_1}(e_2)), \widehat{\gamma_2}(\widehat{\delta_2}(e_1))\,\widehat{}\,\widehat{\gamma_2}(\widehat{\delta_2}(e_2))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

By induction we get that $\Delta; \Gamma \vdash e_1 \sim e_1 : \tau_1 \Rightarrow \tau_2$ and $\Delta; \Gamma \vdash e_2 \sim e_2 : \tau_1$, which tell us that

$$(\widehat{\gamma_1}(\widehat{\delta_1}(e_1)), \widehat{\gamma_2}(\widehat{\delta_2}(e_1))) \in [\![\tau_1 \Rightarrow \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

This implies that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e_1))$. Thus if $\widehat{\gamma}_1(\widehat{\delta}_1(e_1))$ does not terminate, then neither will $\widehat{\gamma}_2(\widehat{\delta}_2(e_1))$ by the above, and thus neither $\widehat{\gamma}_1(\widehat{\delta}_1(e_1\widehat{\ }e_2))$ nor $\widehat{\gamma}_2(\widehat{\delta}_2(e_1\widehat{\ }e_2))$ will by rule $Eccapp_1$. The same can be said about $\widehat{\gamma}_1(\widehat{\delta}_1(e_2))$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_2))$. Thus the result holds in these cases, so we just have to show it holds when all components terminate.

Now suppose that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \mapsto^* f_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \mapsto^* f_2$ for some f_1, f_2, as well as $\widehat{\gamma}_1(\widehat{\delta}_1(e_2)) \mapsto^* v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \mapsto^* v_2$ for some v, v'. But by the definition of the extension of the logical relation to terms, we know from the above that $(f_1, f_2) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ and similarly that $(v_1, v_2) \in [\![\tau_1]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. Then we know that $(f_1\widehat{\ }v_1, f_2\widehat{\ }v_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$ by the definition of the logical type relation over closure-converted arrow types. But then $\widehat{\gamma}_1(\widehat{\delta}_1(e_1\widehat{\ }e_2)) \mapsto^* f_1\widehat{\ }v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1\widehat{\ }e_2)) \mapsto^* f_2\widehat{\ }v_2$ by our assumptions above and the *Eccapp* rules. From this we get that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1\widehat{\ }e_2), \widehat{\gamma}_2(\widehat{\delta}_2(e_1\widehat{\ }e_2)) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

as desired, by application of Corollary 7.5.

Case for $Tpair$
Assume that $\Delta; \Gamma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We need to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\langle e_1, e_2 \rangle)), \widehat{\gamma}_2(\widehat{\delta}_2(\langle e_1, e_2 \rangle))) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

this is equivalent to showing that

$$(\langle \widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_1(\widehat{\delta}_1(e_2)) \rangle, \langle \widehat{\gamma}_2(\widehat{\delta}_2(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \rangle) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

By induction we get that $\Delta; \Gamma \vdash e_1 \sim e_1 : \tau_1$ and $\Delta; \Gamma \vdash e_2 \sim e_2 : \tau_2$, which tell us that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))) \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

Then based on the *Epair* rules, clearly $\langle \widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_1(\widehat{\delta}_1(e_2)) \rangle \simeq \langle \widehat{\gamma}_2(\widehat{\delta}_2(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \rangle$ since the same is true for the two projections. Suppose that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \mapsto^* v_1$, $\widehat{\gamma}_1(\widehat{\delta}_1(e_2)) \mapsto^* v_2$, $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \mapsto^* v_1'$, and $\widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \mapsto^* v_2'$ for some v_1, v_2, v_1', v_2'. Then since $(v_1, v_1') \in [\![\tau_1]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ and $(v_2, v_2') \in [\![\tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ by induction, we know that $(\langle v_1, v_2 \rangle, \langle v_1', v_2' \rangle) \in [\![\tau_1 \times \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ as the values are the corresponding projections (which we can do by applying Corollary 7.5). But then again by Corollary 7.5 and using the *Epair* rules, we get that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\langle e_1, e_2 \rangle)), \widehat{\gamma}_2(\widehat{\delta}_2(\langle e_1, e_2 \rangle))) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

as desired.

Case for $Tproj$
Assume that $\Delta; \Gamma \vdash \pi_i e : \tau_i$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We need to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\pi_i e)), \widehat{\gamma}_2(\widehat{\delta}_2(\pi_i e))) \in [\![\tau_i]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

this is equivalent to showing that

$$(\pi_i \widehat{\gamma}_1(\widehat{\delta}_1(e)), \pi_i \widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\tau_i]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

By induction we get that $\Delta; \Gamma \vdash e \sim e : \tau_1 \times \tau_2$ and $(\widehat{\gamma}_1(\widehat{\delta}_1(e)), \widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$. Thus $\widehat{\gamma}_1(\widehat{\delta}_1(e)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e))$, which implies that $\widehat{\gamma}_1(\widehat{\delta}_1(\pi_i e)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(\pi_i e))$ by rule $Eproj_1$. Also, if $\widehat{\gamma}_1(\widehat{\delta}_1(e)) \mapsto^* v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e)) \mapsto^* v_2$ for some v_1, v_2, then we know $(v_1, v_2) \in [\![\tau_1 \times \tau_2]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ by induction, and thus by definition $(\pi_i v_1, \pi_i v_2) \in [\![\tau_i]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. Then we can apply Corollary 7.5 to get that $(\pi_i \widehat{\gamma}_1(\widehat{\delta}_1(e)), \pi_i \widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\tau_i]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$, as desired.

Case for *Ttlam*
Assume that $\Delta; \Gamma \vdash \Lambda\alpha.e : \forall\alpha.\tau$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We want to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\Lambda\alpha.e)), \widehat{\gamma}_2(\widehat{\delta}_2(\Lambda\alpha.e))) \in [\![\forall\alpha.\tau]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

To do this, we assume that $\tau_1\ \mathtt{type}, \tau_2\ \mathtt{type}$, and $\mathtt{R} \in \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$ such that $\mathtt{R} = \mathtt{R}^{\mathtt{ST}}$. Define $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}$, $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \tau_1$, and $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \tau_2$. We want to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\Lambda\alpha.e))[\tau_1], \widehat{\gamma}_2(\widehat{\delta}_2(\Lambda\alpha.e))[\tau_2]) \in [\![\forall\alpha.\tau]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

this is equivalent to

$$((\Lambda\alpha.\widehat{\gamma}_1(\widehat{\delta}_1(e)))[\tau_1], (\Lambda\alpha.\widehat{\gamma}_2(\widehat{\delta}_2(e)))[\tau_2]) \in [\![\forall\alpha.\tau]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

as we can alpha-vary α to be a variable not included in the substitutions. By Corollary 7.5, we can equivalently show that

$$([\tau_1/\alpha]\widehat{\gamma}_1(\widehat{\delta}_1(e)), [\tau_2/\alpha]\widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\forall\alpha.\tau]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

By induction we know that $\Delta, \alpha; \Gamma \vdash e \sim e : \tau$. Since $\mathtt{R} = \mathtt{R}^{\mathtt{ST}}$, $\eta' : \delta_1' \leftrightarrow \delta_2'$ holds, and we also know that $\gamma_1 \sim_\Gamma \gamma_2[\eta' : \delta_1' \leftrightarrow \delta_2']$ since the added variable α does not appear in any of the substituted terms. By definition of logical equivalance, this means we know that

$$(\widehat{\gamma}_1(\widehat{\delta}_1'(e)), \widehat{\gamma}_2(\widehat{\delta}_2'(e))) \in [\![\forall\alpha.\tau]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

which is equivalent to the desired result above by just separating out the substitutions for α.

Case for *Ttapp*
Assume that $\Delta; \Gamma \vdash e[\tau_1] : \tau$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We want to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e[\tau_1])), \widehat{\gamma}_2(\widehat{\delta}_2(e[\tau_1]))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

We know that $\Delta; \Gamma \vdash e : \forall\alpha.\tau'$, where $\tau = [\tau_1/\alpha]\tau'$. By induction, we get that $\Delta; \Gamma \vdash e \sim e : \forall\alpha.\tau$, which tells us that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e)), \widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\forall\alpha.\tau']\!]^{\mathtt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

This tells us that $\widehat{\gamma}_1(\widehat{\delta}_1(e)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e))$, and if $\widehat{\gamma}_1(\widehat{\delta}_1(e))$ doesn't terminate, than neither does $\widehat{\gamma}_1(\widehat{\delta}_1(e[\tau_1]))$, and similarly if $\widehat{\gamma}_2(\widehat{\delta}_2(e))$ doesn't terminate, than neither does $\widehat{\gamma}_2(\widehat{\delta}_2(e[\tau_1]))$. Thus if either $\widehat{\gamma}_1(\widehat{\delta}_1(e))$ or $\widehat{\gamma}_2(\widehat{\delta}_2(e))$ doesn't terminate, then neither $\widehat{\gamma}_1(\widehat{\delta}_1(e[\tau']))$ nor $\widehat{\gamma}_2(\widehat{\delta}_2(e[\tau_1]))$ will, so the desired result holds in this case.

Now suppose that $\widehat{\gamma}_1(\widehat{\delta}_1(e[\tau_1])) \mapsto^* v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e[\tau_1])) \mapsto^* v_2$ for some v_1, v_2 (we know that if one terminates, the other must as well). From what we got by induction, we know by the definition of the extension of the logical relation to terms that $(v_1, v_2) \in [\![\forall\alpha.\tau']\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$. Define $\mathtt{R} = [\![\tau_1]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$, $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}$, $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \delta_1(\tau_1)$, and $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \delta_2(\tau_1)$. By the definition of the logical relation we know that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e[\tau_1])), \widehat{\gamma}_2(\widehat{\delta}_2(e[\tau_1]))) \in [\![\tau']\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

By Lemma 8.6, this is equivalent to saying that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e[\tau_1])), \widehat{\gamma}_2(\widehat{\delta}_2(e[\tau_1]))) \in [\![[\tau_1/\alpha]\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

which is equivalent to what we wanted to show, since $\tau = [\tau_1/\alpha]\tau'$.

Case for *Tpack*
Assume that $\Delta;\Gamma \vdash \mathtt{pack}[\tau', e]\,\mathtt{as}\,\exists\alpha.\tau : \exists\alpha.\tau$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We want to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\mathtt{pack}[\tau', e]\,\mathtt{as}\,\exists\alpha.\tau)), \widehat{\gamma}_2(\widehat{\delta}_2(\mathtt{pack}[\tau', e]\,\mathtt{as}\,\exists\alpha.\tau))) \in [\![\exists\alpha.\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

or equivalently

$$(\mathtt{pack}[\widehat{\delta}_1(\tau'), \widehat{\gamma}_1(\widehat{\delta}_1(e))]\,\mathtt{as}\,\widehat{\delta}_1(\exists\alpha.\tau))), \mathtt{pack}[\widehat{\delta}_2(\tau'), \widehat{\gamma}_2(\widehat{\delta}_2(e))]\,\mathtt{as}\,\widehat{\delta}_2(\exists\alpha.\tau)))) \in [\![\exists\alpha.\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

Since we know what the value is, we know what the type in the existential is. Define $\mathtt{R} = [\![\tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, which is ST-closed by Lemma 8.1, as well as $\eta' = \eta\otimes\alpha \hookrightarrow \mathtt{R}$, $\delta_1' = \delta_1\otimes\alpha \hookrightarrow \tau_1$, and $\delta_2' = \delta_2\otimes\alpha \hookrightarrow \tau_1$. Then by the definition of the logical relation we can just show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e)), \widehat{\gamma}_2(\widehat{\delta}_2(e))) \in [\![\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

But by induction we know that $\Delta, \alpha; \Gamma \vdash e \sim e : \tau$, which gets us the result above.

Case for *Tunpack*
Assume that $\Delta;\Gamma \vdash \mathtt{unpack}[\alpha, x] = e_1\,\mathtt{in}\,e_2 : \tau$ and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We want to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\mathtt{unpack}[\alpha, x] = e_1\,\mathtt{in}\,e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(\mathtt{unpack}[\alpha, x] = e_1\,\mathtt{in}\,e_2))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

which, since we can assume that α does not occur in Δ and x does not occur in Γ, is equivalent to $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We want to show that

$$(\mathtt{unpack}[\alpha, x] = \widehat{\gamma}_1(\widehat{\delta}_1(e_1))\,\mathtt{in}\,\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \mathtt{unpack}[\alpha, x] = \widehat{\gamma}_2(\widehat{\delta}_2(e_1))\,\mathtt{in}\,\widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

By induction, we get that $\Delta;\Gamma \vdash e_1 \sim e_1 : \exists\alpha.\tau'$ and $\Delta, \alpha; \Gamma, x : \tau' \vdash e_2 \sim e_2 : \tau$, which tells us that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))) \in [\![\exists\alpha.\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

This tells us that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \simeq \widehat{\gamma}_2(\widehat{\delta}_2(e_1))$, and if $\widehat{\gamma}_1(\widehat{\delta}_1(e_1))$ doesn't terminate, than neither does $\widehat{\gamma}_1(\widehat{\delta}_1(\mathtt{unpack}[\alpha, x] = e_1\,\mathtt{in}\,e_2))$, and similarly if $\widehat{\gamma}_2(\widehat{\delta}_2(e_1))$ doesn't terminate, than neither does $\widehat{\gamma}_2(\widehat{\delta}_2(\mathtt{unpack}[\alpha, x] = e_1\,\mathtt{in}\,e_2))$. Thus if either $\widehat{\gamma}_1(\widehat{\delta}_1(e_1))$ or $\widehat{\gamma}_2(\widehat{\delta}_2(e_1))$ doesn't terminate, then neither $\widehat{\gamma}_1(\widehat{\delta}_1(\mathtt{unpack}[\alpha, x] = e_1\,\mathtt{in}\,e_2))$ nor $\widehat{\gamma}_2(\widehat{\delta}_2(\mathtt{unpack}[\alpha, x] = e_1\,\mathtt{in}\,e_2))$ will, so the desired result holds in this case.

Otherwise, we know that both $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \downarrow$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \downarrow$, so thus there exist some v_1, v_2 such that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \mapsto^* v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \mapsto^* v_2$. By the definition of the extension of the logical relation to terms, we know that $(v_1, v_2) \in [\![\exists\alpha.\tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2}$.

> We will need to use Lemma 7.7 to finish this case. First we will define $\mathtt{R}$ to be $[\![\exists\alpha.\tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, except without the additional ST-closure applied to it that it normally has. Also define $\mathtt{Q} = [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, which is ST-closed by Lemma 8.1, and define the functions
>
> $$g_1 = \lambda y : \exists\alpha.\tau'.\, \mathtt{unpack}[\alpha, x] = y\,\mathtt{in}\,\widehat{\gamma}_1(\widehat{\delta}_1(e_2))$$

$$g_2 = \lambda y : \exists\alpha.\tau'.\,\texttt{unpack}[\alpha, x] = y\,\texttt{in}\,\widehat{\gamma}_2(\widehat{\delta}_2(e_2))$$

Assuming that $(v_1', v_2') \in \texttt{R}$, for Lemma 7.7 we want to show that $(g_1\ v_1', g_2\ v_2') \in \texttt{Q}^{\texttt{E}}$. By Corollary 7.5, this is equivalent to showing that

$$(\texttt{unpack}[\alpha, x] = v_1'\,\texttt{in}\,\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \texttt{unpack}[\alpha, x] = v_2'\,\texttt{in}\,\widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in \texttt{Q}^{\texttt{E}}$$

By the definition of $\texttt{R}$, which is from the definition of the logical relation, we know that there exist $\tau_1\,\texttt{type}$, $\tau_2\,\texttt{type}$, and an ST-closed relation $\texttt{R}' \subseteq \texttt{Val}(\tau_1) \times \texttt{Val}(\tau_2)$ such that $(e_1', e_2') \in [\![\tau']\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$ where $v_1' = \texttt{pack}[\tau_1, e_1']\,\texttt{as}\,\exists\alpha.\tau'$, $v_2' = \texttt{pack}[\tau_2, e_2']\,\texttt{as}\,\exists\alpha.\tau'$, $\eta' = \eta \otimes \alpha \hookrightarrow \texttt{R}'$, $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \tau_1$, and $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \tau_2$.

Therefore we know that

$$\exists\alpha.\tau'.\,\texttt{unpack}[\alpha, x] = v_1'\,\texttt{in}\,\widehat{\gamma}_1(\widehat{\delta}_1(e_2)) \mapsto [\tau_1/\alpha][e_1'/x]\widehat{\gamma}_1(\widehat{\delta}_1(e_2)) = \widehat{\gamma}_1'(\widehat{\delta}_1'(e_2))$$

$$\exists\alpha.\tau'.\,\texttt{unpack}[\alpha, x] = v_2'\,\texttt{in}\,\widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \mapsto [\tau_2/\alpha][e_2'/x]\widehat{\gamma}_2(\widehat{\delta}_2(e_2)) = \widehat{\gamma}_2'(\widehat{\delta}_2'(e_2))$$

where $\gamma_1' = \gamma_1 \otimes x \hookrightarrow e_1'$ and $\gamma_2' = \gamma_2 \otimes x \hookrightarrow e_2'$. We know that $\gamma_1' \sim_{\Gamma, x:\tau'} \gamma_2'[\eta' : \delta_1' \leftrightarrow \delta_2']$ since $(e_1', e_2') \in [\![\tau']\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$ from above. Therefore, by induction we know that

$$(\widehat{\gamma}_1'(\widehat{\delta}_1'(e_2)), \widehat{\gamma}_2'(\widehat{\delta}_2'(e_2))) \in [\![\tau]\!]^{\texttt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

We know that $\Delta \vdash \tau$, so τ does not contain α. Thus $\tau = [\tau''/\alpha]\tau$ for any τ''. Thus we can apply Lemma 8.5, which gives us that

$$(\widehat{\gamma}_1'(\widehat{\delta}_1'(e_2)), \widehat{\gamma}_2'(\widehat{\delta}_2'(e_2))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

Therefore, making use of Corollary 7.5, we have shown that $(g_1\ v_1', g_2\ v_2') \in \texttt{Q}^{\texttt{E}}$, as desired.

Then by Lemma 7.7, we now know that for $\texttt{R}$, $\texttt{Q}$, g_1 and g_2 defined above, if $(v_1, v_2) \in \texttt{R}^{\texttt{ST}}$ then $(g_1\ v_1, g_2\ v_2) \in \texttt{Q}^{\texttt{E}}$. We already know that $(v_1, v_2) \in [\![\exists\alpha.\tau']\!] = \texttt{R}^{\texttt{ST}}$ by definition, so therefore $(g_1\ v_1, g_2\ v_2) \in \texttt{Q}^{\texttt{E}}$. By Lemma 7.5, this is equivalent to saying that

$$(\texttt{unpack}[\alpha, x] = v_1\,\texttt{in}\,\widehat{\gamma}_1(\widehat{\delta}_1(e_2)), \texttt{unpack}[\alpha, x] = v_2\,\texttt{in}\,\widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

Since we know that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \mapsto^* v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \mapsto^* v_2$, by rule *Eunpack*$_1$ we can again apply Lemma 7.5 to get the desired result, that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\texttt{unpack}[\alpha, x] = e_1\,\texttt{in}\,e_2)), \widehat{\gamma}_2(\widehat{\delta}_2(\texttt{unpack}[\alpha, x] = e_1\,\texttt{in}\,e_2))) \in [\![\tau]\!]^{\texttt{E}}_{\eta:\delta_1 \leftrightarrow \delta_2}$$

□

9.2 Congruence

Lemma 9.2. If $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\Delta'; \Gamma' \triangleright \tau')$ and $\Delta; \Gamma \vdash e_1 \sim e_2 : \tau$, then $\Delta'; \Gamma' \vdash \mathcal{C}\{e_1\} \sim \mathcal{C}\{e_2\} : \tau'$.

Proof. First, define the function $f = \lambda x : \tau.\mathcal{C}\{x\}$. By our definition of contexts, clearly $\Delta'; \Gamma' \vdash f : \tau \to \tau'$. Assume that $\gamma \sim_{\Gamma'} \gamma'[\eta : \delta_1 \leftrightarrow \delta_2]$. We want to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\mathcal{C}\{e_1\})), \widehat{\gamma}_2(\widehat{\delta}_2(\mathcal{C}\{e_2\}))) \in [\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

but by our definition of f and using Corollary 7.5, this is the same as showing that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(f\ e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(f\ e_2))) \in [\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

which is also the same as showing that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(f))\ \widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(f))\ \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

By Theorem 9.1, we know that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(f)), \widehat{\gamma}_2(\widehat{\delta}_2(f))) \in [\![\tau \to \tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

and thus clearly $(\widehat{\gamma}_1(\widehat{\delta}_1(f)), \widehat{\gamma}_2(\widehat{\delta}_2(f))) \in [\![\tau \to \tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ since f is a value. By assumption, we know that $(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. We consider two cases:

If $\widehat{\gamma}_1(\widehat{\delta}_1(e_1))$ does not terminate, then neither does $\widehat{\gamma}_2(\widehat{\delta}_2(e_2))$. Clearly, then, neither do $\widehat{\gamma}_1(\widehat{\delta}_1(f))\ \widehat{\gamma}_1(\widehat{\delta}_1(e_1))$ nor $\widehat{\gamma}_2(\widehat{\delta}_2(f))\ \widehat{\gamma}_2(\widehat{\delta}_2(e_2))$, by rule Eapp_2. This implies the desired result.

If $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \downarrow$, then also $\widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \downarrow$. This means that there exist some v_1, v_2 such that $\widehat{\gamma}_1(\widehat{\delta}_1(e_1)) \mapsto^* v_1$ and $\widehat{\gamma}_2(\widehat{\delta}_2(e_2)) \mapsto^* v_2$. Also, since we know that $(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$, we know that $(v_1, v_2) \in [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. However, by the definition of $[\![\tau \to \tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, this means that $(\widehat{\gamma}_1(\widehat{\delta}_1(f))\ v_1, \widehat{\gamma}_2(\widehat{\delta}_2(f))\ v_2) \in [\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. We can then apply Corollary 7.5 to get that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(f))\ \widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(f))\ \widehat{\gamma}_2(\widehat{\delta}_2(e_2))) \in [\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

which is what we wanted to show. □

9.3 Respect for Contextual Equivalence

Lemma 9.3. Suppose that $(e_1, e_2) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. If $e_1' \cong_{\widehat{\delta}_1(\tau)} e_1$ and $e_2' \cong_{\widehat{\delta}_2(\tau)} e_2$, then $(e_1', e_2') \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$.

Proof. We want to show that $(e_1', e_2') \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. We first have to show that $e_1' \simeq e_2'$. By Lemma 5.1 and the assumption that $e_1' \cong_{\widehat{\delta}_1(\tau)} e_1$, we get that $e_1' \simeq e_1$. Similarly, we get that $e_2 \simeq e_2'$. We also get that $e_1 \simeq e_2$ by the assumption that $(e_1, e_2) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. Thus by transitivity of Kleene equivalence, we have that $e_1' \simeq e_2'$, as desired. Now we just have to show that if $e_1' \mapsto^* v_1'$ and $e_2' \mapsto^* v_2'$ for some v_1', v_2', then $(v_1', v_2') \in [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. By the above it must then be the case that $e_1 \mapsto^* v_1$ and $e_2 \mapsto^* v_2$ for some v_1, v_2 where $(v_1, v_2) \in [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$.

Suppose $(f_1, f_2) \in [\![\tau]\!]^{\mathtt{S}}_{\eta:\delta_1\leftrightarrow\delta_2}$. By Lemma 5.7, we know that $v_1' \cong_{\widehat{\delta}_1(\tau)} e_1'$ and $e_1 \cong_{\widehat{\delta}_1(\tau)} v_1$, and thus by transitivity $v_1' \cong_{\widehat{\delta}_1(\tau)} v_1$. Using the context $\mathcal{C}_1 = f_1\ o$, we then get by Corollary 5.2 that $\mathcal{C}_1\{v_1'\} \simeq \mathcal{C}\{v_1\}$, or equivalently $f_1\ v_1' \simeq f_1\ v_1$. By a similar method using the context $\mathcal{C}_2 = f_2\ o$, we get that $f_2\ v_2 \simeq f_2\ v_2'$. Also, by the definition of the S operation, we know that $f_1\ v_1 \simeq f_2\ v_2$ since $(v_1, v_2) \in [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$. Then by transitivity we get that $f_1\ v_1' \simeq f_2\ v_2'$. However, since f_1 and f_2 were arbitrary, this holds for all f_1, f_2, and thus we have that $(v_1', v_2') \in [\![\tau]\!]^{\mathtt{ST}}_{\eta:\delta_1\leftrightarrow\delta_2}$ by definition. However, by Lemma 8.1, this means that $(v_1', v_2') \in [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$, as desired. ☐

9.4 Logical Equivalence implies Contextual Equivalence

Theorem 9.4. If $\Delta;\Gamma \vdash e_1 \sim e_2 : \tau$, then $\Delta;\Gamma \vdash e_1 \cong e_2 : \tau$.

Proof. Assume that $\Delta;\Gamma \vdash e_1 \sim e_2 : \tau$. Let $\mathcal{C} : (\Delta;\Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \mathtt{int})$ be a program context. By Lemma 9.2, we know that $\cdot \vdash \mathcal{C}\{e_1\} \sim \mathcal{C}\{e_2\} : \mathtt{int}$. Since the context is empty, we get that $(\mathcal{C}\{e_1\}, \mathcal{C}\{e_2\}) \in [\![\mathtt{int}]\!]^{\mathtt{E}}$, which by definition implies that $\mathcal{C}\{e_1\} \simeq \mathcal{C}\{e_2\}$. Then by the definition of contextual equivalence, $\Delta;\Gamma \vdash e_1 \cong e_2 : \tau$. ☐

9.5 Contextual Equivalence implies Logical Equivalence

Theorem 9.5. If $\Delta;\Gamma \vdash e_1 \cong e_2 : \tau$, then $\Delta;\Gamma \vdash e_1 \sim e_2 : \tau$.

Proof. Assume that $\Delta;\Gamma \vdash e_1 \cong e_2 : \tau$ and that $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$. We want to show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

We know that $\Delta;\Gamma \vdash e_1 \cong e_1 : \tau$. By repeated application of Lemma 5.3 to $\Delta;\Gamma \vdash e_1 \cong e_2 : \tau$, we get that

$$\cdot; \widehat{\delta}_2(\Gamma) \vdash \widehat{\delta}_2(e_1) \cong \widehat{\delta}_2(e_2) : \widehat{\delta}_2(\tau)$$

Then define γ_2' such that $\gamma_2'(x) = \widehat{\delta}_2(\gamma_2(x))$ for all $x \in \Gamma$. Clearly then $\gamma_2' : \widehat{\delta}_2(\Gamma)$, so by Lemma 5.4, we get that

$$\widehat{\delta}_2(\widehat{\gamma}_2'(e_1)) \cong_{\widehat{\delta}_2(\tau)} \widehat{\delta}_2(\widehat{\gamma}_2'(e_2))$$

which is equivalent to

$$\widehat{\gamma}_2(\widehat{\delta}_2(e_1)) \cong_{\widehat{\delta}_2(\tau)} \widehat{\gamma}_2(\widehat{\delta}_2(e_2))$$

By Lemma 9.1, we know that $\Delta;\Gamma \vdash e_1 \cong e_1 : \tau$, which tells us that $(\widehat{\gamma}_1(\widehat{\delta}_1(e_1)), \widehat{\gamma}_2(\widehat{\delta}_2(e_1))) \in [\![\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$. The result follows from this and the above derivation by applying Lemma 9.3. ☐

Theorem 9.6. $\Delta;\Gamma \vdash e_1 \cong e_2 : \tau$ iff $\Delta;\Gamma \vdash e_1 \sim e_2 : \tau$.

Proof. By Theorem 9.4 and Theorem 9.5. ☐

10 Closure Conversion

10.1 Translation

We define the standard closure conversion type translation $|\tau|$ and term translation $\Delta;\Gamma \vdash_S e : \tau \rightsquigarrow \bar{e}$.

The type translation is defined as follows:

$$
\begin{aligned}
|\alpha| &= \alpha \\
|\texttt{unit}| &= \texttt{unit} \\
|\texttt{int}| &= \texttt{int} \\
|\tau_1 \times \tau_2| &= |\tau_1| \times |\tau_2| \\
|\tau_1 \to \tau_2| &= \exists\alpha.((|\tau_1| \times \alpha) \Rightarrow |\tau_2|) \times \alpha \\
|\forall\alpha.\tau| &= \forall\alpha.|\tau| \\
|\exists\alpha.\tau| &= \exists\alpha.|\tau|
\end{aligned}
$$

We also define $|\Gamma|$ by $|\cdot| = \cdot$ and $|\Gamma, x : \tau| = |\Gamma|, x : |\tau|$

The term translation is defined as follows:

$$\frac{}{\Delta;\Gamma \vdash_S \texttt{()} : \texttt{unit} \rightsquigarrow \texttt{()}}\ \textit{Runit} \qquad \frac{\Gamma(x) = \tau}{\Delta;\Gamma \vdash_S x : \tau \rightsquigarrow x}\ \textit{Rvar}$$

$$\frac{}{\Delta;\Gamma \vdash_S n : \texttt{int} \rightsquigarrow n}\ \textit{Rint} \qquad \frac{\Delta;\Gamma \vdash_S e_1 : \texttt{int} \rightsquigarrow \bar{e_1} \quad \Delta;\Gamma \vdash_S e_2 : \texttt{int} \rightsquigarrow \bar{e_2}}{\Delta;\Gamma \vdash_S e_1\,\texttt{p}\,e_2 : \texttt{int} \rightsquigarrow \bar{e_1}\,\texttt{p}\,\bar{e_2}}\ \textit{Rintop}$$

$$\frac{\Delta;\Gamma \vdash_S e_1 : \texttt{int} \rightsquigarrow \bar{e_1} \quad \Delta;\Gamma \vdash_S e_2 : \tau \rightsquigarrow \bar{e_2} \quad \Delta;\Gamma \vdash_S e_3 : \tau \rightsquigarrow \bar{e_3}}{\Delta;\Gamma \vdash_S \texttt{ifz}(e_1, e_2, e_3) : \tau \rightsquigarrow \texttt{ifz}(\bar{e_1}, \bar{e_2}, \bar{e_3})}\ \textit{Rifz}$$

$$\frac{\Delta;\Gamma \vdash_S e_1 : \tau_1 \rightsquigarrow \bar{e_1} \quad \Delta;\Gamma \vdash_S e_2 : \tau_2 \rightsquigarrow \bar{e_2}}{\Delta;\Gamma \vdash_S \langle e_1, e_2\rangle : \tau_1 \times \tau_2 \rightsquigarrow \langle \bar{e_1}, \bar{e_2}\rangle}\ \textit{Rpair} \qquad \frac{\Delta;\Gamma \vdash_S e : \tau_1 \times \tau_2 \rightsquigarrow \bar{e} \quad i \in \{1,2\}}{\Delta;\Gamma \vdash_S \pi_i e : \tau_i \rightsquigarrow \pi_i \bar{e}}\ \textit{Rproj}$$

$$\frac{\Delta,\alpha;\Gamma \vdash_S e : \tau \rightsquigarrow \bar{e}}{\Delta;\Gamma \vdash_S \Lambda\alpha.e : \forall\alpha.\tau \rightsquigarrow \Lambda\alpha.\bar{e}}\ \textit{Rtlam} \qquad \frac{\Delta;\Gamma \vdash_S e : \forall\alpha.\tau \rightsquigarrow \bar{e} \quad \Delta \vdash_S \tau'\ \texttt{type}}{\Delta;\Gamma \vdash_S e[\tau'] : [\tau'/\alpha]\tau \rightsquigarrow \bar{e}[|\tau'|]}\ \textit{Rtapp}$$

$$\frac{\Delta \vdash_S \tau'\ \texttt{type} \quad \Delta,\alpha \vdash_S \tau\ \texttt{type} \quad \Delta;\Gamma \vdash_S e : [\tau'/\alpha]\tau \rightsquigarrow \bar{e}}{\Delta;\Gamma \vdash_S \texttt{pack}[\tau', e]\ \texttt{as}\ \exists\alpha.\tau : \exists\alpha.\tau \rightsquigarrow \texttt{pack}[|\tau'|, \bar{e}]\ \texttt{as}\ \exists\alpha.|\tau|}\ \textit{Rpack}$$

$$\frac{\Delta;\Gamma \vdash_S e_1 : \exists\alpha.\tau_1 \rightsquigarrow \bar{e_1} \quad \Delta,\alpha;\Gamma, x : \tau_1 \vdash_S e_2 : \tau_2 \rightsquigarrow \bar{e_2} \quad \Delta \vdash_S \tau_2\ \texttt{type}}{\Delta;\Gamma \vdash_S \texttt{unpack}[\alpha, x] = e_1\ \texttt{in}\ e_2 : \tau_2 \rightsquigarrow \texttt{unpack}[\alpha, x] = \bar{e_1}\ \texttt{in}\ \bar{e_2}}\ \textit{Runpack}$$

$$\frac{\begin{array}{c}\Gamma = x_1 : \tau_1, ..., x_n : \tau_n \qquad \Delta \vdash_S \tau\,\texttt{type} \\ \Delta; \Gamma, x : \tau, f : \tau \to \tau' \vdash_S e : \tau' \rightsquigarrow \bar{e} \quad \tau_{env} = |\tau_1| \times ... \times |\tau_n|\end{array}}{\begin{array}{l}\Delta; \Gamma \vdash_S \texttt{fun}\ f(x : \tau).e : \tau \to \tau' \rightsquigarrow \texttt{pack}[\tau_{env}, \\ \quad \langle(\widehat{\texttt{fun}}\ f(y : |\tau| \times \tau_{env}).[\texttt{pack}[\tau_{env}, \langle f, \pi_2 y\rangle]\ \texttt{as}\ |\tau \to \tau'|/f] \\ \qquad [\pi_1 y/x][\pi_1\pi_2 y/x_1]...[\pi_1\pi_2...\pi_2 y/x_{n-1}][\pi_2...\pi_2 y/x_n]\bar{e} \\ \quad), \langle x_1, \langle ... \langle x_{n-1}, x_n\rangle ...\rangle\rangle\rangle]\ \texttt{as}\ |\tau \to \tau'|\end{array}}\ Rfun$$

$$\frac{\Delta; \Gamma \vdash_S e_1 : \tau \to \tau' \rightsquigarrow \bar{e_1} \quad \Delta; \Gamma \vdash_S e_2 : \tau \rightsquigarrow \bar{e_2}}{\Delta; \Gamma \vdash_S e_1\ e_2 : \tau' \rightsquigarrow \texttt{unpack}[\alpha, x] = \bar{e_1}\ \texttt{in}(\pi_1 x)\hat{}\langle \bar{e_2}, \pi_2 x\rangle}\ Rapp$$

10.2 Fix vs. Recursive Functions

There is a reason that we didn't use the general `fix` for recursion. Using `fix` instead of recurisve functions results in problems when doing closure conversion. What follows is an example showing the problem that arises when using `fix` instead of inherently recursive functions. First, the static and dynamic rules for `fix` would be:

$$\frac{\Delta; \Gamma, x : \tau \vdash e : \tau}{\Delta; \Gamma \vdash \texttt{fix}\ (x : \tau).e : \tau}\ Tfix$$

$$\frac{}{\texttt{fix}\ (x : \tau).e \mapsto [\texttt{fix}\ (x : \tau).e/x]e}\ Efix$$

Now consider the term

$$\texttt{fix}\ (f : \texttt{int} \to \texttt{int}).\lambda x : \texttt{int}.\texttt{ifz}(x, 0, f\ (x - 1))$$

This doesn't really do anything interesting, but will demonstrate the problem. After closure conversion, this translates to (assuming no context)

$$\texttt{fix}\ (f : \texttt{int} \to \texttt{int}).\,\texttt{pack}[\texttt{int} \to \texttt{int}, \langle\widehat{\lambda} y : \texttt{int} \times (\texttt{int} \to \texttt{int}).\texttt{ifz}(\pi_1 y, 0, E), f\rangle$$

Where $E = \texttt{unpack}[\alpha, z] = \pi_2 y\,\texttt{in}(\pi_1 z)\hat{}\langle \pi_1 y - 1, \pi_2 z\rangle$. Define the above to be F. Then this will step to

$$\texttt{pack}[\texttt{int} \to \texttt{int}, \langle\widehat{\lambda} y : \texttt{int} \times (\texttt{int} \to \texttt{int}).\texttt{ifz}(\pi_1 y, 0, E), F\rangle$$

This is because the whole term gets substituted in for f due to the definition of `fix` . Call the above F'. However, for this to be a value, we need the inner term to also be a value (that is, the term $\langle\widehat{\lambda} y : \texttt{int} \times (\texttt{int} \to \texttt{int}).\texttt{ifz}(\pi_1 y, 0, E), F\rangle$). But we just said that F steps to F', so the second term in the pair can make a step to F'. But wait, we just said that F' can make a step, as the second term in the pair in the pack can step. Thus this can step, and so on. Thus we never actually get to a value, hence why `fix` was not used with closure conversion and recursive functions were used instead.

11 Language Conversion

11.1 Over and Back

We define two functions in the combined language based on types in the languages. The first is $\texttt{over}_\tau : \tau \to |\tau|$, which takes terms in the source language and messes with them at the top level to get their type to line up with the second language. The second is $\texttt{back}_\tau : |\tau| \to \tau$, which does the opposite, taking terms in the target language and messing with them at the top level to get their type to line up with the source language. The two functions are mutually recursive and are defined as follows:

$$
\begin{aligned}
\texttt{over}_\alpha &= \lambda x : \alpha . x \\
\texttt{over}_{\texttt{unit}} &= \lambda x : \texttt{unit}. x \\
\texttt{over}_{\texttt{int}} &= \lambda x : \texttt{int}. x \\
\texttt{over}_{\tau_1 \times \tau_2} &= \lambda x : \tau_1 \times \tau_2 . \langle \texttt{over}_{\tau_1} \pi_1 x, \texttt{over}_{\tau_2} \pi_2 x \rangle \\
\texttt{over}_{\tau_1 \to \tau_2} &= \lambda f : \tau_1 \to \tau_2 . \, \texttt{pack}[\tau_1 \to \tau_2, \langle \widehat{\lambda} y : |\tau_1| \times \tau_1 \to \tau_2 . \\
&\qquad \texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), f \rangle] \, \texttt{as} \, |\tau_1 \to \tau_2| \\
\texttt{over}_{\forall \alpha . \tau} &= \lambda x : (\forall \alpha . \tau) . \Lambda \alpha . (\texttt{over}_\tau (x[\alpha])) \\
\texttt{over}_{\exists \alpha . \tau} &= \lambda x : (\exists \alpha . \tau) . \, \texttt{unpack}[\alpha, y] = x \, \texttt{in} (\texttt{pack}[\alpha, \texttt{over}_\tau (y)] \, \texttt{as} \, |\exists \alpha . \tau|)
\end{aligned}
$$

$$
\begin{aligned}
\texttt{back}_\alpha &= \lambda x : \alpha . x \\
\texttt{back}_{\texttt{unit}} &= \lambda x : \texttt{unit}. x \\
\texttt{back}_{\texttt{int}} &= \lambda x : \texttt{int}. x \\
\texttt{back}_{\tau_1 \times \tau_2} &= \lambda x : |\tau_1 \times \tau_2| . \langle \texttt{back}_{\tau_1} \pi_1 x, \texttt{back}_{\tau_2} \pi_2 x \rangle \\
\texttt{back}_{\tau_1 \to \tau_2} &= \lambda f : |\tau_1 \to \tau_2| . \lambda y : \tau_1 . \, \texttt{unpack}[\alpha, g] = f \, \texttt{in} \, \texttt{back}_{\tau_2} ((\pi_1 g) \hat{} \langle \texttt{over}_{\tau_1} y, \pi_2 g \rangle) \\
\texttt{back}_{\forall \alpha . \tau} &= \lambda x : |\forall \alpha . \tau| . \Lambda \alpha . (\texttt{back}_\tau (x[\alpha])) \\
\texttt{back}_{\exists \alpha . \tau} &= \lambda x : |\exists \alpha . \tau| . \, \texttt{unpack}[\alpha, y] = x \, \texttt{in} (\texttt{pack}[\alpha, \texttt{back}_\tau (y)] \, \texttt{as} \, \exists \alpha . \tau)
\end{aligned}
$$

11.2 Inverses

Lemma 11.1. For all $\Delta; \Gamma \vdash e : \tau$ and $\Delta; |\Gamma| \vdash e' : \tau$,

$$\Delta; \Gamma \vdash \texttt{back}_\tau (\texttt{over}_\tau \ e) \cong e : \tau$$

$$\Delta; |\Gamma| \vdash \texttt{over}_\tau (\texttt{back}_\tau \ e') \cong e' : |\tau|$$

Proof. By induction on the structure of τ.

If e does not halt, then clearly neither does $\texttt{back}_\tau(\texttt{over}_\tau \ e)$, so clearly

$$\Delta; \Gamma \vdash \texttt{back}_\tau (\texttt{over}_\tau \ e) \cong e : \tau$$

If $e \downarrow$, then $e \mapsto^* v$ for some $v \ \texttt{val}$. Then by the cases below we'll have that $\Delta; \Gamma \vdash \texttt{back}_\tau(\texttt{over}_\tau \ v) \cong v : \tau$ from which the desired result will follow from Lemma 5.7. We can do similarly with e', which if it does not halt we have the desired result, and if instead $e' \mapsto^* v'$ for $v' \ \texttt{val}$, we can apply the below and Lemma 5.7 to get the desired result.

Case for $\tau = \alpha$
In this case we want to show that

$$\Delta; \Gamma \vdash \mathtt{back}_\alpha(\mathtt{over}_\alpha\ v) \cong v : \alpha$$

which is equivalent to

$$\Delta; \Gamma \vdash (\lambda x : \alpha.x)\ ((\lambda x : \alpha.x)\ v) \cong v : \alpha$$

However we know that by our assumption that $\Delta; \Gamma \vdash v : \alpha$ and Lemma 5.7,

$$\Delta; \Gamma \vdash (\lambda x : \alpha.x)\ ((\lambda x : \alpha.x)\ v) \cong (\lambda x : \alpha.x)\ v : \alpha$$

$$\Delta; \Gamma \vdash (\lambda x : \alpha.x)\ v \cong v : \alpha$$

The desired result follows from transitivity. Since $\mathtt{back}_\alpha = \mathtt{over}_\alpha$, the reverse inverse is identical in this case.

Case for $\tau = \mathtt{unit}$
In this case we want to show that

$$\Delta; \Gamma \vdash \mathtt{back}_{\mathtt{unit}}(\mathtt{over}_{\mathtt{unit}}\ v) \cong v : \mathtt{unit}$$

which is equivalent to

$$\Delta; \Gamma \vdash (\lambda x : \mathtt{unit}.x)\ ((\lambda x : \mathtt{unit}.x)\ v) \cong v : \mathtt{unit}$$

However we know that by our assumption that $\Delta; \Gamma \vdash v : \mathtt{unit}$ and Lemma 5.7,

$$\Delta; \Gamma \vdash (\lambda x : \mathtt{unit}.x)\ ((\lambda x : \mathtt{unit}.x)\ v) \cong (\lambda x : \mathtt{unit}.x)\ v : \mathtt{unit}$$

$$\Delta; \Gamma \vdash (\lambda x : \mathtt{unit}.x)\ v \cong v : \mathtt{unit}$$

The desired result follows from transitivity. Since $\mathtt{back}_{\mathtt{unit}} = \mathtt{over}_{\mathtt{unit}}$, the reverse inverse is identical in this case.

Case for $\tau = \mathtt{int}$
In this case we want to show that

$$\Delta; \Gamma \vdash \mathtt{back}_{\mathtt{int}}(\mathtt{over}_{\mathtt{int}}\ v) \cong v : \mathtt{int}$$

which is equivalent to

$$\Delta; \Gamma \vdash (\lambda x : \mathtt{int}.x)\ ((\lambda x : \mathtt{int}.x)\ v) \cong v : \mathtt{int}$$

However we know that by our assumption that $\Delta; \Gamma \vdash v : \mathtt{int}$ and Lemma 5.7,

$$\Delta; \Gamma \vdash (\lambda x : \mathtt{int}.x)\ ((\lambda x : \mathtt{int}.x)\ v) \cong (\lambda x : \mathtt{int}.x)\ v : \mathtt{int}$$

$$\Delta; \Gamma \vdash (\lambda x : \mathtt{int}.x)\ v \cong v : \mathtt{int}$$

The desired result follows from transitivity. Since $\mathtt{back}_{\mathtt{int}} = \mathtt{over}_{\mathtt{int}}$, the reverse inverse is identical in this case.

Case for $\tau = \tau_1 \times \tau_2$
In this case we want to show that

$$\Delta; \Gamma \vdash \texttt{back}_{\tau_1 \times \tau_2}(\texttt{over}_{\tau_1 \times \tau_2}\ v) \cong v : \tau_1 \times \tau_2$$

which is equivalent to

$$\Delta; \Gamma \vdash (\lambda x : |\tau|.\langle \texttt{back}_{\tau_1} \pi_1 x, \texttt{back}_{\tau_2} \pi_2 x \rangle)\ ((\lambda x : \tau.\langle \texttt{over}_{\tau_1} \pi_1 x, \texttt{over}_{\tau_2} \pi_2 x \rangle)\ v) \cong v : \tau$$

However we know that by our assumption that $\Delta; \Gamma \vdash v : \tau_1 \times \tau_2$, so by rule $Tproj$ that $\Delta; \Gamma \vdash \pi_1 v : \tau_1$ and $\Delta; \Gamma \vdash \pi_2 v : \tau_2$, so by induction we have that

$$\Delta; \Gamma \vdash \texttt{back}_{\tau_1}(\texttt{over}_{\tau_1} \pi_1 v) \cong \pi_1 v : \tau_1$$

$$\Delta; \Gamma \vdash \texttt{back}_{\tau_2}(\texttt{over}_{\tau_2} \pi_2 v) \cong \pi_2 v : \tau_2$$

Then using Lemma 5.7 and the above,

$$\begin{aligned} \Delta; \Gamma \vdash\ & (\lambda x : |\tau|.\langle \texttt{back}_{\tau_1} \pi_1 x, \texttt{back}_{\tau_2} \pi_2 x \rangle)\ ((\lambda x : \tau.\langle \texttt{over}_{\tau_1} \pi_1 x, \texttt{over}_{\tau_2} \pi_2 x \rangle)\ v) \\ & \cong (\lambda x : |\tau|.\langle \texttt{back}_{\tau_1} \pi_1 x, \texttt{back}_{\tau_2} \pi_2 x \rangle)\ (\langle \texttt{over}_{\tau_1} \pi_1 v, \texttt{over}_{\tau_2} \pi_2 v \rangle) \\ & \cong \langle \texttt{back}_{\tau_1} \pi_1 \langle \texttt{over}_{\tau_1} \pi_1 v, \texttt{over}_{\tau_2} \pi_2 v \rangle, \texttt{back}_{\tau_2} \pi_2 \langle \texttt{over}_{\tau_1} \pi_1 v, \texttt{over}_{\tau_2} \pi_2 v \rangle \rangle \\ & \cong \langle \texttt{back}_{\tau_1}(\texttt{over}_{\tau_1} \pi_1 v), \texttt{back}_{\tau_2}(\texttt{over}_{\tau_2} \pi_2 v) \rangle \\ & \cong \langle \pi_1 v, \pi_2 v \rangle \\ & \cong v \\ & : \tau_1 \times \tau_2 \end{aligned}$$

The desired result follows from transitivity. The reverse direction carries out in much the same way.

Case for $\tau = \tau_1 \to \tau_2$
In this case we want to show that

$$\Delta; \Gamma \vdash \texttt{back}_{\tau_1 \to \tau_2}(\texttt{over}_{\tau_1 \to \tau_2}\ v) \cong v : \tau_1 \to \tau_2$$

which is equivalent to

$$\begin{aligned} \Delta; \Gamma \vdash\ & (\lambda f : |\tau|.\lambda y : \tau_1.\,\texttt{unpack}[\alpha, g] = f\ \texttt{in}\,\texttt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \texttt{over}_{\tau_1} y, \pi_2 g \rangle)) \\ & \quad ((\lambda f : \tau.\,\texttt{pack}[\texttt{unit}, \langle \widehat{\lambda} y : |\tau_1| \times \texttt{unit}.\texttt{over}_{\tau_2}(f\ (\texttt{back}_{\tau_1} \pi_1 y)), \texttt{()} \rangle]\ \texttt{as}\ |\tau|)\ v) \\ & \cong v : \tau \end{aligned}$$

We get by induction that for all $\Delta; \Gamma \vdash e_1 : \tau_1$ and $\Delta; \Gamma \vdash e_2 : \tau_2$,

$$\Delta; \Gamma \vdash \texttt{back}_{\tau_1}(\texttt{over}_{\tau_1} e_1) \cong e_1 : \tau_1$$

$$\Delta; \Gamma \vdash \texttt{back}_{\tau_2}(\texttt{over}_{\tau_2} e_2) \cong e_2 : \tau_2$$

Thus, using Lemma 5.7 the fact that $\Delta;\Gamma \vdash v : \tau_1 \to \tau_2$, we can show that

$$\begin{aligned}
\Delta;\Gamma \vdash\ & (\lambda f : |\tau|.\lambda z : \tau_1.\, \mathtt{unpack}[\alpha, g] = f\, \mathtt{in}\, \mathtt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{\tau_1} z, \pi_2 g\rangle)) \\
&\quad ((\lambda f : \tau.\, \mathtt{pack}[\mathtt{unit}, \langle \widehat{\lambda} y : |\tau_1| \times \mathtt{unit}.\mathtt{over}_{\tau_2}(f\ (\mathtt{back}_{\tau_1}\pi_1 y)), \mathtt{()}\rangle]\, \mathtt{as}\, |\tau|)\ v) \\
\cong\ & (\lambda f : |\tau|.\lambda z : \tau_1.\, \mathtt{unpack}[\alpha, g] = f\, \mathtt{in}\, \mathtt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{\tau_1} z, \pi_2 g\rangle)) \\
&\quad (\mathtt{pack}[\mathtt{unit}, \langle \widehat{\lambda} y : |\tau_1| \times \mathtt{unit}.\mathtt{over}_{\tau_2}(v\ (\mathtt{back}_{\tau_1}\pi_1 y)), \mathtt{()}\rangle]\, \mathtt{as}\, |\tau|) \\
\cong\ & \lambda z : \tau_1.\, \mathtt{unpack}[\alpha, g] = (\mathtt{pack}[\mathtt{unit}, \langle \widehat{\lambda} y : |\tau_1| \times \mathtt{unit}.\mathtt{over}_{\tau_2}(v\ (\mathtt{back}_{\tau_1}\pi_1 y)), \mathtt{()}\rangle]\, \mathtt{as}\, |\tau|) \\
&\quad \mathtt{in}\, \mathtt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{\tau_1} z, \pi_2 g\rangle) \\
\cong\ & \lambda z : \tau_1.\mathtt{back}_{\tau_2}((\pi_1\langle \widehat{\lambda} y : |\tau_1| \times \mathtt{unit}.\mathtt{over}_{\tau_2}(v\ (\mathtt{back}_{\tau_1}\pi_1 y)), \mathtt{()}\rangle)\hat{} \\
&\quad \langle \mathtt{over}_{\tau_1} z, \pi_2\langle \widehat{\lambda} y : |\tau_1| \times \mathtt{unit}.\mathtt{over}_{\tau_2}(v\ (\mathtt{back}_{\tau_1}\pi_1 y)), \mathtt{()}\rangle\rangle) \\
\cong\ & \lambda z : \tau_1.\mathtt{back}_{\tau_2}((\widehat{\lambda} y : |\tau_1| \times \mathtt{unit}.\mathtt{over}_{\tau_2}(v\ (\mathtt{back}_{\tau_1}\pi_1 y)))\hat{}\langle \mathtt{over}_{\tau_1} z, \mathtt{()}\rangle) \\
\cong\ & \lambda z : \tau_1.\mathtt{back}_{\tau_2}(\mathtt{over}_{\tau_2}(v\ (\mathtt{back}_{\tau_1}\pi_1\langle \mathtt{over}_{\tau_1} z, \mathtt{()}\rangle))) \\
\cong\ & \lambda z : \tau_1.\mathtt{back}_{\tau_2}(\mathtt{over}_{\tau_2}(v\ (\mathtt{back}_{\tau_1}(\mathtt{over}_{\tau_1} z)))) \\
\cong\ & \lambda z : \tau_1.\mathtt{back}_{\tau_2}(\mathtt{over}_{\tau_2}(v\ z)) \\
\cong\ & \lambda z : \tau_1.v\ z \\
\cong\ & v : \tau
\end{aligned}$$

So the desired result follows from transitivity.

Now we want to show the opposite direction, that

$$\Delta; |\Gamma| \vdash \mathtt{over}_{\tau_1\to\tau_2}(\mathtt{back}_{\tau_1\to\tau_2}\ v') \cong v' : |\tau_1 \to \tau_2|$$

which is equivalent to

$$\begin{aligned}
\Delta; |\Gamma| \vdash\ & (\lambda f : \tau.\, \mathtt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\mathtt{over}_{\tau_2}((\pi_2 y)\ (\mathtt{back}_{\tau_1}\pi_1 y)), f\rangle]\, \mathtt{as}\, |\tau|) \\
&\quad ((\lambda f : |\tau|.\lambda y : \tau_1.\, \mathtt{unpack}[\alpha, g] = f\, \mathtt{in}\, \mathtt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{\tau_1} y, \pi_2 g\rangle))\ v') \\
\cong\ & v' : |\tau|
\end{aligned}$$

We get by induction that for all $\Delta; |\Gamma| \vdash e_1 : |\tau_1|$ and $\Delta; |\Gamma| \vdash e_2 : |\tau_2|$,

$$\Delta; |\Gamma| \vdash \mathtt{over}_{\tau_1}(\mathtt{back}_{\tau_1} e_1) \cong e_1 : |\tau_1|$$

$$\Delta; |\Gamma| \vdash \mathtt{over}_{\tau_2}(\mathtt{back}_{\tau_2} e_2) \cong e_2 : |\tau_2|$$

Since v' is a value, we can assume that $v' = \mathtt{pack}[\tau', \langle h, e\rangle]\, \mathtt{as}\, |\tau_1 \to \tau_2|$ for an appropriate τ', h, e, where $h\, \mathtt{val}$ and $e\, \mathtt{val}$. Thus, using Lemma 5.7 and the fact that $\Delta; |\Gamma| \vdash v' : |\tau_1 \to \tau_2|$, we can show that

$$\Delta; |\Gamma| \vdash (\lambda f : \tau.\, \texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), f\rangle]\, \texttt{as}\, |\tau|)$$
$$((\lambda f : |\tau|.\lambda y : \tau_1.\, \texttt{unpack}[\alpha, g] = f\, \texttt{in}\, \texttt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \texttt{over}_{\tau_1} y, \pi_2 g\rangle))\ v')$$
$$\cong (\lambda f : \tau.\, \texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), f\rangle]\, \texttt{as}\, |\tau|)$$
$$(\lambda y : \tau_1.\, \texttt{unpack}[\alpha, g] = v'\, \texttt{in}\, \texttt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \texttt{over}_{\tau_1} y, \pi_2 g\rangle))$$
$$\cong \texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), \lambda y : \tau_1.$$
$$\texttt{unpack}[\alpha, g] = v'\, \texttt{in}\, \texttt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \texttt{over}_{\tau_1} y, \pi_2 g\rangle)\rangle]\, \texttt{as}\, |\tau|$$
$$\cong \texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), \lambda y : \tau_1.$$
$$\texttt{unpack}[\alpha, g] = (\texttt{pack}[\tau', \langle h, e\rangle]\, \texttt{as}\, |\tau|)\, \texttt{in}\, \texttt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \texttt{over}_{\tau_1} y, \pi_2 g\rangle)\rangle]\, \texttt{as}\, |\tau|$$
$$\cong \texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), \lambda y : \tau_1.$$
$$\texttt{back}_{\tau_2}((\pi_1 \langle h, e\rangle)\hat{}\langle \texttt{over}_{\tau_1} y, \pi_2 \langle h, e\rangle\rangle)\rangle]\, \texttt{as}\, |\tau|$$
$$\cong \texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), \lambda y : \tau_1.$$
$$\texttt{back}_{\tau_2}(h\hat{}\langle \texttt{over}_{\tau_1} y, e\rangle)\rangle]\, \texttt{as}\, |\tau| : |\tau|$$

For conciseness, define $F = \lambda y : \tau_1.\texttt{back}_{\tau_2}(h\hat{}\langle \texttt{over}_{\tau_1} y, e\rangle)$. Thus all that we need to show is

$$\Delta; |\Gamma| \vdash \texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), F\rangle]\, \texttt{as}\, |\tau| \cong \texttt{pack}[\tau', \langle h, e\rangle]\, \texttt{as}\, |\tau| : |\tau|$$

From which the desired result will follow, since $v' = \texttt{pack}[\tau', \langle h, e\rangle]\, \texttt{as}\, |\tau|$. To show the above, we will first use the coincidence of logical and contextual equivalence to instead show that

$$\Delta; |\Gamma| \vdash \texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), F\rangle]\, \texttt{as}\, |\tau| \sim \texttt{pack}[\tau', \langle h, e\rangle]\, \texttt{as}\, |\tau| : |\tau|$$

Which means that we must show that for any $\delta_1 : \Delta$, $\delta_2 : \Delta$, and $\eta : \delta_1 \leftrightarrow \delta_2$, with $\gamma_1 \sim_{|\Gamma|} \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$, that

$$(\widehat{\delta}_1(\widehat{\gamma}_1(\texttt{pack}[\tau, \langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), F\rangle]\, \texttt{as}\, |\tau|)), \widehat{\delta}_2(\widehat{\gamma}_2(\texttt{pack}[\tau', \langle h, e\rangle]\, \texttt{as}\, |\tau|)))$$
$$\in [\![\exists \alpha.((|\tau_1| \times \alpha) \Rightarrow |\tau_2|) \times \alpha]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$$

By definition, this is the case if there exists some ST-closed $\texttt{R} \subseteq \texttt{Val}(\tau) \times \texttt{Val}(\widehat{\delta}_2(\tau'))$ such that

$$(\langle \widehat{\delta}_1(\widehat{\gamma}_1(\langle \widehat{\lambda} y : |\tau_1| \times \tau.\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1} \pi_1 y)), F\rangle)), \texttt{()}\rangle, \langle \widehat{\delta}_2(\widehat{\gamma}_2(h)), \widehat{\delta}_2(\widehat{\gamma}_2(e))\rangle)$$
$$\in [\![((|\tau_1| \times \alpha) \Rightarrow |\tau_2|) \times \alpha]\!]_{\eta':\delta'_1 \leftrightarrow \delta'_2}$$

Where $\delta'_1 = \delta_1 \otimes \alpha \hookrightarrow \tau_1 \to \tau_2$, $\delta'_2 = \delta_2 \otimes \alpha \hookrightarrow \widehat{\delta}_2(\tau')$, and $\eta' = \eta \otimes \alpha \hookrightarrow \texttt{R}$. So just define $\texttt{R} = \{(e_1, e_2) \mid (\widehat{\delta}_1(\widehat{\gamma}_1(F)), e_1) \in [\![\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ and $(\widehat{\delta}_2(\widehat{\gamma}_2(e)), e_2) \in [\![\tau']\!]_{\eta:\delta_1 \leftrightarrow \delta_2}\}$, which is clearly ST-closed. Then we just need to show that each of the projections are valid. By the definition of $\texttt{R}$, we already know that $(\widehat{\delta}_1(\widehat{\gamma}_1(F)), \widehat{\delta}_2(\widehat{\gamma}_2(e))) \in [\![\alpha]\!]_{\eta':\delta'_1 \leftrightarrow \delta'_2}$, so all that we have left to show is that

$$(\widehat{\delta}_1(\widehat{\gamma}_1(\widehat{\lambda} x : |\tau_1| \times (\tau_1 \to \tau_2).(h\hat{}\langle \pi_1 x, e\rangle))), \widehat{\delta}_2(\widehat{\gamma}_2(h))) \in [\![(|\tau_1| \times \alpha) \Rightarrow |\tau_2|]\!]_{\eta':\delta'_1 \leftrightarrow \delta'_2}$$

To show this we assume that $(v_1, v_2) \in [\![|\tau_1| \times \alpha]\!]_{\eta':\delta'_1 \leftrightarrow \delta'_2}$ and show that

$$(\widehat{\delta}_1(\widehat{\gamma}_1(\widehat{\lambda} x : |\tau_1| \times (\tau_1 \to \tau_2).(h\hat{}\langle \pi_1 x, e\rangle)))\hat{}v_1, \widehat{\delta}_2(\widehat{\gamma}_2(h))\hat{}v_2) \in [\![|\tau_2|]\!]^{\texttt{E}}_{\eta':\delta'_1 \leftrightarrow \delta'_2}$$

Which by Lemma 7.5 is the same as showing

$$(\widehat{\delta}_1(\widehat{\gamma}_1(h\hat{}\langle \pi_1 v_1, e\rangle)), \widehat{\delta}_2(\widehat{\gamma}_2(h\hat{}v_2))) \in [\![|\tau_2|]\!]^{\texttt{E}}_{\eta':\delta'_1 \leftrightarrow \delta'_2}$$

Since we know by reflexivity that

$$(\widehat{\delta}_1(\widehat{\gamma}_1(h)), \widehat{\delta}_2(\widehat{\gamma}_2(h))) \in [\![(|\tau_1| \times \tau') \Rightarrow |\tau_2|]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$$

By definition we then just have to show that

$$(\widehat{\delta}_1(\widehat{\gamma}_1(\langle \pi_1 v_1, e\rangle)), \widehat{\delta}_2(\widehat{\gamma}_2(v_2))) \in [\![|\tau_1| \times \tau']\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

We know by assumption that

$$(\widehat{\delta}_1(\widehat{\gamma}_1(\pi_1 v_1)), \widehat{\delta}_2(\widehat{\gamma}_2(\pi_1 v_2))) \in [\![|\tau_1|]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

And by the definition of $\mathtt{R}$, it follows that

$$(\widehat{\delta}_1(\widehat{\gamma}_1(e)), \widehat{\delta}_2(\widehat{\gamma}_2(\pi_2 v_2))) \in [\![\tau']\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

So the desired result follows.

Case for $\tau = \forall\alpha.\tau'$
In this case we want to show that

$$\Delta;\Gamma \vdash \mathtt{back}_{\forall\alpha.\tau'}(\mathtt{over}_{\forall\alpha.\tau'}\ v) \cong v : \forall\alpha.\tau'$$

which is equivalent to

$$\Delta;\Gamma \vdash (\lambda x : |\tau|.\Lambda\alpha.\mathtt{back}_{\tau'}(x[\alpha]))\ ((\lambda x : \tau.\Lambda\alpha.\mathtt{over}_{\tau'}(x[\alpha]))\ v) \cong v : \forall\alpha.\tau'$$

However we know that by our assumption that $\Delta;\Gamma \vdash v : \forall\alpha.\tau'$, so by rule *Ttapp* that $\Delta;\Gamma \vdash v[\tau''] : [\tau''/\alpha]\tau$, so by induction we have that

$$\Delta;\Gamma \vdash \mathtt{back}_{\tau'}(\mathtt{over}_{\tau'}(v[\tau''])) \cong v[\tau''] : [\tau''/\alpha]\tau'$$

Then using Lemma 5.7 and the above,

$$\begin{aligned}
\Delta;\Gamma \vdash\ & (\lambda x : |\tau|.\Lambda\alpha.\mathtt{back}_{\tau'}(x[\alpha]))\ ((\lambda x : \tau.\Lambda\alpha.\mathtt{over}_{\tau'}(x[\alpha]))\ v) \\
& \cong (\lambda x : |\tau|.\Lambda\alpha.\mathtt{back}_{\tau'}(x[\alpha]))\ (\Lambda\alpha.\mathtt{over}_{\tau'}(v[\alpha])) \\
& \cong \Lambda\alpha.\mathtt{back}_{\tau'}((\Lambda\alpha.\mathtt{over}_{\tau'}(v[\alpha]))[\alpha]) \\
& \cong \Lambda\alpha.\mathtt{back}_{\tau'}(\mathtt{over}_{\tau'}(v[\alpha])) \\
& \cong \Lambda\alpha.v[\alpha] \\
& \cong v \\
& : \forall\alpha.\tau'
\end{aligned}$$

The desired result follows from transitivity. The reverse direction carries out in much the same way.

Case for $\tau = \exists\alpha.\tau'$
In this case we want to show that

$$\Delta;\Gamma \vdash \mathtt{back}_{\exists\alpha.\tau'}(\mathtt{over}_{\exists\alpha.\tau'}\ v) \cong v : \exists\alpha.\tau'$$

which is equivalent to

$$\begin{aligned}
\Delta;\Gamma \vdash & (\lambda x : |\tau|.\, \mathtt{unpack}[\alpha, y] = x\, \mathtt{in}(\mathtt{pack}[\alpha, \mathtt{back}_{\tau'} y]\, \mathtt{as}\, \tau)) \\
& ((\lambda x : \tau.\, \mathtt{unpack}[\alpha, y] = x\, \mathtt{in}(\mathtt{pack}[\alpha, \mathtt{over}_{\tau'} y]\, \mathtt{as}\, |\tau|))\ v) \cong v : \exists\alpha.\tau'
\end{aligned}$$

Using the fact that $\Delta;\Gamma \vdash v : \exists\alpha.\tau'$ and it is a value, we know that $v = \texttt{pack}[\tau'', e]\,\texttt{as}\,\exists\alpha.\tau'$ for some τ'', e such that $e\,\texttt{val}$ by rule $Tpack$. We get by induction that for all $\Delta, \alpha; \Gamma \vdash e : \tau'$,

$$\Delta;\Gamma \vdash \texttt{back}_{\tau'}(\texttt{over}_{\tau'} e) \cong e : \tau'$$

Thus using Lemma 5.7, we can show that

$$\begin{aligned}
\Delta;\Gamma \vdash\ & (\lambda x : |\tau|.\,\texttt{unpack}[\alpha, y] = x\,\texttt{in}(\texttt{pack}[\alpha, \texttt{back}_{\tau'} y]\,\texttt{as}\,\tau)) \\
&\quad ((\lambda x : \tau.\,\texttt{unpack}[\alpha, y] = x\,\texttt{in}(\texttt{pack}[\alpha, \texttt{over}_{\tau'} y]\,\texttt{as}\,|\tau|))\ v) \\
&\cong (\lambda x : |\tau|.\,\texttt{unpack}[\alpha, y] = x\,\texttt{in}(\texttt{pack}[\alpha, \texttt{back}_{\tau'} y]\,\texttt{as}\,\tau)) \\
&\quad (\texttt{unpack}[\alpha, y] = v\,\texttt{in}(\texttt{pack}[\alpha, \texttt{over}_{\tau'} y]\,\texttt{as}\,|\tau|)) \\
&\cong (\lambda x : |\tau|.\,\texttt{unpack}[\alpha, y] = x\,\texttt{in}(\texttt{pack}[\alpha, \texttt{back}_{\tau'} y]\,\texttt{as}\,\tau)) \\
&\quad (\texttt{unpack}[\alpha, y] = (\texttt{pack}[\tau'', e]\,\texttt{as}\,\exists\alpha.\tau')\,\texttt{in}(\texttt{pack}[\alpha, \texttt{over}_{\tau'} y]\,\texttt{as}\,|\tau|)) \\
&\cong (\lambda x : |\tau|.\,\texttt{unpack}[\alpha, y] = x\,\texttt{in}(\texttt{pack}[\alpha, \texttt{back}_{\tau'} y]\,\texttt{as}\,\tau))\ (\texttt{pack}[\tau'', \texttt{over}_{\tau'} e]\,\texttt{as}\,|\tau|) \\
&\cong \texttt{unpack}[\alpha, y] = (\texttt{pack}[\tau'', \texttt{over}_{\tau'} e]\,\texttt{as}\,|\tau|)\,\texttt{in}(\texttt{pack}[\alpha, \texttt{back}_{\tau'} y]\,\texttt{as}\,\tau) \\
&\cong \texttt{pack}[\tau'', ([\tau''/\alpha]\texttt{back}_{\tau'})((([\tau''/\alpha]\texttt{over}_{\tau'})e)]\,\texttt{as}\,\tau \\
&\cong \texttt{pack}[\tau'', e]\,\texttt{as}\,\tau \\
&\cong v \\
&: \exists\alpha.\tau'
\end{aligned}$$

The desired result follows from transitivity. The reverse direction carries out in much the same way.

□

Lemma 11.2. The functions $\texttt{over}_\tau$ and $\texttt{back}_\tau$ are inverses of one another. That is,

$$\Delta;\Gamma \vdash \texttt{back}_\tau \circ \texttt{over}_\tau \cong id : \tau \to \tau$$

$$\Delta;|\Gamma| \vdash \texttt{over}_\tau \circ \texttt{back}_\tau \cong id : |\tau| \to |\tau|$$

Proof. By the coincidence of contextual and logical equivalence, we can equivalently show that

$$\Delta;\Gamma \vdash \texttt{back}_\tau \circ \texttt{over}_\tau \sim id : \tau \to \tau$$

Let $\gamma_1 : \Gamma$, $\gamma_2 : \Gamma$, $\delta_1 : \Delta$, $\delta_2 : \Delta$, and $\eta : \delta_1 \leftrightarrow \delta_2$. Thus we can equivalently show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\texttt{back}_\tau \circ \texttt{over}_\tau)), \widehat{\gamma}_2(\widehat{\delta}_2(\lambda x : \tau.x))) \in [\![\tau \to \tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$$

which by the definition of function composition is the same as

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\lambda x : \tau.(\texttt{back}_\tau(\texttt{over}_\tau\ x)))), \widehat{\gamma}_2(\widehat{\delta}_2(\lambda x : \tau.x))) \in [\![\tau \to \tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$$

To show this, we assume that $(v_1, v_2) \in [\![\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$ and show that

$$(\widehat{\gamma}_1(\widehat{\delta}_1(\lambda x : \tau.(\texttt{back}_\tau(\texttt{over}_\tau\ x))))\ v_1, \widehat{\gamma}_2(\widehat{\delta}_2(\lambda x : \tau.x))\ v_2) \in [\![\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$$

which is equivalent to

$$(\widehat{\gamma}_1(\widehat{\delta}_1((\lambda x : \tau.(\texttt{back}_\tau(\texttt{over}_\tau\ x)))\ v_1)), \widehat{\gamma}_2(\widehat{\delta}_2((\lambda x : \tau.x)\ v_2))) \in [\![\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$$

However, by Lemma 11.1 we know that

$$\widehat{\gamma}_1(\widehat{\delta}_1((\lambda x:\tau.(\texttt{back}_\tau(\texttt{over}_\tau\ x)))\ v_1)) \cong_{\widehat{\delta}_1(\tau)} \widehat{\gamma}_1(\widehat{\delta}_1(\texttt{back}_\tau(\texttt{over}_\tau\ v_1))) \cong_{\widehat{\delta}_1(\tau)} \widehat{\gamma}_1(\widehat{\delta}_1(v_1)) \cong_{\widehat{\delta}_1(\tau)} v_1 : \tau$$

$$\widehat{\gamma}_2(\widehat{\delta}_2((\lambda x:\tau.x)\ v_2)) \cong_{\widehat{\delta}_2(\tau)} \widehat{\gamma}_2(\widehat{\delta}_2(v_2)) \cong_{\widehat{\delta}_2(\tau)} v_2 : \tau$$

So by Lemma 9.3 this is the same as shoing that $(v_1, v_2) \in [\![\tau]\!]_{\eta:\delta_1 \leftrightarrow \delta_2}$, however we assumed this already, so the desired result follows. $\square$

11.3 The Back Relation

Lemma 11.3. If $\Delta, \alpha \vdash_S \tau\,\texttt{type}$ and $\Delta \vdash_S \tau'\,\texttt{type}$, then for any $\delta_1 : \Delta$, $\delta_2 : \Delta$, and $\eta : \delta_1 \leftrightarrow \delta_2$, there exists an ST-closed relation $\mathtt{R} \in \mathtt{Val}(\widehat{\delta}_1(\tau')) \times \mathtt{Val}(\widehat{\delta}_2(|\tau'|))$ defined as

$$\mathtt{R} = \{(v_1, v_2) \,|\, (v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})(v_2)) \in [\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\}$$

such that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_\tau), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau})) \in [\![|\tau| \to [\tau'/\alpha]\tau]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{over}_\tau), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau})) \in [\![[\tau'/\alpha]\tau \to |\tau|]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

Where $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}$, $\delta'_1 = \delta_1 \otimes \alpha \hookrightarrow \widehat{\delta}_1(\tau')$, $\delta'_2 = \delta_2 \otimes \alpha \hookrightarrow \widehat{\delta}_2(|\tau'|)$, We know that $\mathtt{R}$ is ST-closed because it is defined in terms of $[\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$, which is itself ST-closed.

Proof. By induction on the structure of τ.

Case for $\tau = \mathtt{unit}$
In this case we want to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\mathtt{unit}}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\mathtt{unit}})) \in [\![|\mathtt{unit}| \to [\tau'/\alpha]\mathtt{unit}]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

This is equivalent to

$$(\mathtt{back}_{\mathtt{unit}}, \mathtt{back}_{\mathtt{unit}}) \in [\![\mathtt{unit} \to \mathtt{unit}]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

This follows immediately from Reflexivity. The case for `over` works exactly the same.

Case for $\tau = \mathtt{int}$
In this case we want to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\mathtt{int}}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\mathtt{int}})) \in [\![|\mathtt{int}| \to [\tau'/\alpha]\mathtt{int}]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

This is equivalent to

$$(\mathtt{back}_{\mathtt{int}}, \mathtt{back}_{\mathtt{int}}) \in [\![\mathtt{int} \to \mathtt{int}]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

This follows immediately from Reflexivity. The case for `over` works exactly the same.

Case for $\tau = \alpha$
In this case we want to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_\alpha), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\alpha})) \in [\![|\alpha| \to [\tau'/\alpha]\alpha]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

This is equivalent to

$$(\widehat{\delta}_1([\tau'/\alpha](\lambda x : \alpha.x)), \widehat{\delta}_2(\mathtt{back}_{\tau'})) \in [\![\alpha \to \tau']\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

And similarly equivalent to

$$(\lambda x : \widehat{\delta}_1(\tau').x, \widehat{\delta}_2(\mathtt{back}_{\tau'})) \in [\![\alpha \to \tau']\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

To show this we assume that $(v_1, v_2) \in [\![\alpha]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$, which implies that $(v_1, v_2) \in \mathtt{R}$, and show that

$$((\lambda x : \widehat{\delta}_1(\tau').x)\ v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2) \in [\![\tau']\!]^{\mathtt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

However, since τ' doesn't reference α, we can equivalently show that

$$(v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2) \in [\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

This follows immediately from our definition of $\mathtt{R}$.

Now we need to show the other half, that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{over}_\alpha), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\alpha})) \in [\![[\tau'/\alpha]\alpha \to |\alpha|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This is equivalent to

$$(\widehat{\delta}_1([\tau'/\alpha](\lambda x : \alpha.x)), \widehat{\delta}_2(\mathtt{over}_{\tau'})) \in [\![\tau' \to \alpha]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

And similarly equivalent to

$$(\lambda x : \widehat{\delta}_1(\tau').x, \widehat{\delta}_2(\mathtt{over}_{\tau'})) \in [\![\tau' \to \alpha]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

To show this we assume that $(v_1, v_2) \in [\![\tau']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and show that

$$((\lambda x : \widehat{\delta}_1(\tau').x)\ v_1, \widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2) \in [\![\alpha]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This is equivalent to showing that

$$(v_1, \widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2) \in [\![\alpha]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Which is the same as showing that $(v_1, \widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2) \in \mathtt{R}^{\mathtt{E}}$. We know by Lemma 11.1 that

$$v_2 \cong_{\widehat{\delta}_2(\tau')} \widehat{\delta}_2(\mathtt{back}_{\tau'})(\widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2)$$

So by Lemma 9.3, we know that from our assumption that

$$(v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})(\widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2)) \in [\![\tau']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Since $v_1 \downarrow$, this implies that $\widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2 \downarrow$, so we have that $\widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2 \mapsto^* v_2'$ for some value v_2'. Then we know by Lemma 7.5 that

$$(v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2') \in [\![\tau']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

But by definition this implies that $(v_1, v_2') \in \mathtt{R}$, and therefore that $(v_1, \widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2) \in \mathtt{R}^{\mathtt{E}}$.

Case for $\tau = \tau_1 \times \tau_2$
In this case we want to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\tau_1\times\tau_2}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha](\tau_1\times\tau_2)})) \in [\![|\tau_1 \times \tau_2| \to [\tau'/\alpha](\tau_1 \times \tau_2)]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

To show this we assume that $(v_1, v_2) \in [\![|\tau_1 \times \tau_2|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and show

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\tau_1\times\tau_2})\ v_1, \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha](\tau_1\times\tau_2)})\ v_2) \in [\![[\tau'/\alpha](\tau_1 \times \tau_2)]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of back, this is the same as showing

$$\begin{aligned}&(\widehat{\delta}_1(([\tau'/\alpha](\lambda x : |\tau_1 \times \tau_2|.\langle \mathtt{back}_{\tau_1}(\pi_1 x), \mathtt{back}_{\tau_2}(\pi_2 x)\rangle))(v_1)),\\ &\widehat{\delta}_2(\lambda x : |[\tau'/\alpha](\tau_1 \times \tau_2)|.\langle \mathtt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 x), \mathtt{back}_{[\tau'/\alpha]\tau_2}(\pi_2 x)\rangle(v_2))) \in [\![[\tau'/\alpha](\tau_1 \times \tau_2)]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

By Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\langle([\tau'/\alpha]\mathtt{back}_{\tau_1})(\pi_1 v_1), ([\tau'/\alpha]\mathtt{back}_{\tau_2})(\pi_2 v_1)\rangle),\\ &\widehat{\delta}_2(\langle\mathtt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2), \mathtt{back}_{[\tau'/\alpha]\tau_2}(\pi_2 v_2)\rangle)) \in [\![[\tau'/\alpha]\tau_1 \times [\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then using Lemma 8.3 for pairs we just need to show that

$$(\widehat{\delta}_1(([\tau'/\alpha]\mathtt{back}_{\tau_1})(\pi_1 v_1)), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))) \in [\![[\tau'/\alpha]\tau_1]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1(([\tau'/\alpha]\mathtt{back}_{\tau_2})(\pi_2 v_1)), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_2}(\pi_2 v_2))) \in [\![[\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

However by induction we get that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\tau_1}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_1})) \in [\![|\tau_1| \to [\tau'/\alpha]\tau_1]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\tau_2}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_2})) \in [\![|\tau_2| \to [\tau'/\alpha]\tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We also know by our assumption and the definition of the type relation that $(\pi_1 v_1, \pi_1 v_2) \in [\![|\tau_1|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$ and $(\pi_2 v_1, \pi_2 v_2) \in [\![|\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$, from which the desired result follows by Lemma 8.3 for functions.

Now we want to show this for `over` as well, so we want to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{over}_{\tau_1\times\tau_2}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha](\tau_1\times\tau_2)})) \in [\![[\tau'/\alpha](\tau_1 \times \tau_2) \to |\tau_1 \times \tau_2|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

To show this we assume that $(v_1, v_2) \in [\![[\tau'/\alpha](\tau_1 \times \tau_2)]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and show

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{over}_{\tau_1\times\tau_2})\ v_1, \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha](\tau_1\times\tau_2)})\ v_2) \in [\![|\tau_1 \times \tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of `over`, this is the same as showing

$$\begin{aligned}&(\widehat{\delta}_1([\tau'/\alpha]\lambda x : \tau_1 \times \tau_2.\langle\mathtt{over}_{\tau_1}(\pi_1 x), \mathtt{over}_{\tau_2}(\pi_2 x)\rangle\ v_1),\\ &\widehat{\delta}_2(\lambda x : [\tau'/\alpha](\tau_1 \times \tau_2).\langle\mathtt{over}_{[\tau'/\alpha]\tau_1}(\pi_1 x), \mathtt{over}_{[\tau'/\alpha]\tau_2}(\pi_2 x)\rangle\ v_2)) \in [\![|\tau_1 \times \tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

By Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\langle([\tau'/\alpha]\mathtt{over}_{\tau_1})(\pi_1 v_1), ([\tau'/\alpha]\mathtt{over}_{\tau_2})(\pi_2 v_1)\rangle),\\ &\widehat{\delta}_2(\langle\mathtt{over}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2), \mathtt{over}_{[\tau'/\alpha]\tau_2}(\pi_2 v_2)\rangle)) \in [\![|\tau_1 \times \tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then using Lemma 8.3 for pairs we just need to show that

$$(\widehat{\delta}_1(([\tau'/\alpha]\mathtt{over}_{\tau_1})(\pi_1 v_1)), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))) \in [\![|\tau_1|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1(([\tau'/\alpha]\mathtt{over}_{\tau_2})(\pi_2 v_1)), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_2}(\pi_2 v_2))) \in [\![|\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

However by induction we get that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{over}_{\tau_1}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_1})) \in [\![[\tau'/\alpha]\tau_1 \to |\tau_1|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{over}_{\tau_2}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_2})) \in [\![[\tau'/\alpha]\tau_2 \to |\tau_2|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We also know by our assumption and the definition of the type relation that $(\pi_1 v_1, \pi_1 v_2) \in [\![[\tau'/\alpha]\tau_1]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$ and $(\pi_2 v_1, \pi_2 v_2) \in [\![[\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$, from which the desired result follows by Lemma 8.3 for functions.

Case for $\tau = \tau_1 \to \tau_2$
In this case we want to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\tau_1\to\tau_2}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha](\tau_1\to\tau_2)})) \in [\![|\tau_1 \to \tau_2| \to [\tau'/\alpha](\tau_1 \to \tau_2)]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We assume that

$$(f_1, f_2) \in [\![|\tau_1 \to \tau_2|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'} = [\![\exists\beta.(|\tau_1| \times \beta \Rightarrow |\tau_2|) \times \beta]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

So by definition we just need to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\tau_1\to\tau_2})\ f_1, \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_1\to[\tau'/\alpha]\tau_2})\ f_2) \in [\![[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

However, this no longer depends on α, so we can just show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\tau_1\to\tau_2}\ f_1), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_1\to[\tau'/\alpha]\tau_2}\ f_2)) \in [\![[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

By the definition of back this is the same as showing

$$\begin{aligned}&(\widehat{\delta}_1(([\tau'/\alpha](\lambda f : |\tau_1 \to \tau_2|.\lambda y : \tau_1.\,\mathtt{unpack}[\beta, g] = f\,\mathtt{in}\,\mathtt{back}_{\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{\tau_1} y, \pi_2 g\rangle)))\ f_1),\\ &\widehat{\delta}_2((\lambda f : |[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2|.\lambda y : [\tau'/\alpha]\tau_1.\,\mathtt{unpack}[\beta, g] = f\,\mathtt{in}\\ &\quad \mathtt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{[\tau'/\alpha]\tau_1} y, \pi_2 g\rangle))\ f_2)) \in [\![[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1((\lambda y : [\tau'/\alpha]\tau_1.\,\mathtt{unpack}[\beta, g] = f_1\,\mathtt{in}([\tau'/\alpha]\mathtt{back}_{\tau_2})((\pi_1 g)\hat{}\langle([\tau'/\alpha]\mathtt{over}_{\tau_1})y, \pi_2 g\rangle))),\\ &\widehat{\delta}_2(\lambda y : [\tau'/\alpha]\tau_1.\,\mathtt{unpack}[\beta, g] = f_2\,\mathtt{in}\,\mathtt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{[\tau'/\alpha]\tau_1} y, \pi_2 g\rangle)))\\ &\quad \in [\![[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2]\!]_{\eta:\delta_1\leftrightarrow\delta_2}\end{aligned}$$

To show this we assume that $(v_1, v_2) \in [\![[\tau'/\alpha]\tau_1]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and show that

$$\begin{aligned}&(\widehat{\delta}_1(\lambda y : [\tau'/\alpha]\tau_1.\,\mathtt{unpack}[\beta, g] = f_1\,\mathtt{in}([\tau'/\alpha]\mathtt{back}_{\tau_2})((\pi_1 g)\hat{}\langle([\tau'/\alpha]\mathtt{over}_{\tau_1})y, \pi_2 g\rangle))\ v_1,\\ &\widehat{\delta}_2(\lambda y : [\tau'/\alpha]\tau_1.\,\mathtt{unpack}[\beta, g] = f_2\,\mathtt{in}\,\mathtt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{[\tau'/\alpha]\tau_1} y, \pi_2 g\rangle))\ v_2)\\ &\quad \in [\![[\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\mathtt{unpack}[\beta, g] = f_1\,\mathtt{in}([\tau'/\alpha]\mathtt{back}_{\tau_2})((\pi_1 g)\hat{}\langle([\tau'/\alpha]\mathtt{over}_{\tau_1})v_1, \pi_2 g\rangle)),\\ &\widehat{\delta}_2(\mathtt{unpack}[\beta, g] = f_2\,\mathtt{in}\,\mathtt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g\rangle)))\\ &\quad \in [\![[\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\end{aligned}$$

Since f_1 is a value, we know by rule *Vpack* that it is of the form $\mathtt{pack}[\tau_1', g_1]\,\mathtt{as}\,|\tau_1 \to \tau_2|$ for appropriate type τ_1' and value g_1. Similarly we can write f_2 as $\mathtt{pack}[\tau_2', g_2]\,\mathtt{as}\,|\tau_1 \to \tau_2|$. Thus we can show that

$$\begin{aligned}&(\widehat{\delta}_1(\mathtt{unpack}[\beta, g] = \mathtt{pack}[\tau_1', g_1]\,\mathtt{as}\,|\tau_1 \to \tau_2|\,\mathtt{in}([\tau'/\alpha]\mathtt{back}_{\tau_2})((\pi_1 g)\hat{}\langle([\tau'/\alpha]\mathtt{over}_{\tau_1})v_1, \pi_2 g\rangle)),\\ &\widehat{\delta}_2(\mathtt{unpack}[\beta, g] = \mathtt{pack}[\tau_2', g_2]\,\mathtt{as}\,|\tau_1 \to \tau_2|\,\mathtt{in}\,\mathtt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle \mathtt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g\rangle)))\\ &\quad \in [\![[\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\end{aligned}$$

Thus by Lemma 7.5, we can just show that

$$(\widehat{\delta}_1(([\tau'/\alpha]\texttt{back}_{\tau_2})((\pi_1 g_1)\widehat{\ }\langle([\tau'/\alpha]\texttt{over}_{\tau_1})v_1, \pi_2 g_1\rangle)),$$
$$\widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g_2)\widehat{\ }\langle\texttt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g_2\rangle))) \in [\![[\tau'/\alpha]\tau_2]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

However since we know by induction that

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{back}_{\tau_2}), \widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha]\tau_2})) \in [\![|\tau_2| \to [\tau'/\alpha]\tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

So by Lemma 8.3 we just have to show that

$$\widehat{\delta}_1((\pi_1 g_1)\widehat{\ }\langle([\tau'/\alpha]\texttt{over}_{\tau_1})v_1, \pi_2 g_1\rangle), \widehat{\delta}_2((\pi_1 g_2)\widehat{\ }\langle\texttt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g_2\rangle)) \in [\![|\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Now by our assumption that $(f_1, f_2) \in [\![|\tau_1 \to \tau_2|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, we know that there exists a relation $\mathtt{R}' \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$ such that

$$(g_1, g_2) \in [\![(|\tau_1| \times \beta \Rightarrow |\tau_2|) \times \beta]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

Where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1'$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2'$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \mathtt{R}'$. This tells us that by definition,

$$(\pi_1 g_1, \pi_1 g_2) \in [\![|\tau_1| \times \beta \Rightarrow |\tau_2|]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

$$(\pi_2 g_1, \pi_2 g_2) \in [\![\beta]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

We also know by induction that

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\tau_1}), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau_1})) \in [\![[\tau'/\alpha]\tau_1 \to |\tau_1|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Which by definition tells us that

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\tau_1})\ v_1, \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau_1})\ v_2) \in [\![|\tau_1|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Which then by Lemma 8.3 for pairs we know that

$$(\widehat{\delta}_1(\langle[\tau'/\alpha]\texttt{over}_{\tau_1})\ v_1, \pi_2 g_1\rangle, \widehat{\delta}_2(\langle\texttt{over}_{[\tau'/\alpha]\tau_1})\ v_2, \pi_2 g_1\rangle) \in [\![|\tau_1| \times \beta]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

Then again by Lemma 8.3 for functions we know that

$$\widehat{\delta}_1((\pi_1 g_1)\widehat{\ }\langle([\tau'/\alpha]\texttt{over}_{\tau_1})v_1, \pi_2 g_1\rangle), \widehat{\delta}_2((\pi_1 g_2)\widehat{\ }\langle\texttt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g_2\rangle)) \in [\![|\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Which is what we wanted to show.

Now we need to show the corresponding case for `over`, that is

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\tau_1\to\tau_2}), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha](\tau_1\to\tau_2)})) \in [\![[\tau'/\alpha](\tau_1 \to \tau_2) \to |\tau_1 \to \tau_2|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We assume that

$$(f_1, f_2) \in [\![[\tau'/\alpha](\tau_1 \to \tau_2)]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

So by definition we just need to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\tau_1\to\tau_2})\ f_1, \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau_1\to[\tau'/\alpha]\tau_2})\ f_2) \in [\![|\tau_1 \to \tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of over this is the same as showing

$$(\widehat{\delta}_1(([\tau'/\alpha](\lambda f : \tau_1 \to \tau_2.\, \texttt{pack}[\tau_1 \to \tau_2, \langle\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2).\\ \texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1}(\pi_1 y))), f\rangle]\,\texttt{as}\,|\tau_1 \to \tau_2|))\ f_1),\\ \widehat{\delta}_2((\lambda f : [\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2.\, \texttt{pack}[\tau_1 \to \tau_2, \langle\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\\ \texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y))), f\rangle]\,\texttt{as}\,|[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2|)\ f_2)) \in [\![|\tau_1 \to \tau_2|]\!]^{\texttt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

Then by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1([\tau'/\alpha]\,\texttt{pack}[\tau_1 \to \tau_2, \langle\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1}(\pi_1 y))), f_1\rangle]\,\texttt{as}\,|\tau_1 \to \tau_2|),\\ \widehat{\delta}_2(\texttt{pack}[\tau_1 \to \tau_2, \langle\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y))), f_2\rangle]\\ \texttt{as}\,|[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2|)) \in [\![\exists\beta.(|\tau_1| \times \beta \Rightarrow |\tau_2|) \times \beta]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

To show this, we pick $\texttt{R}' = [\![[\tau'/\alpha](\tau_1 \to \tau_2)]\!]$ and use the definition of the type relation to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\langle\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1}(\pi_1 y))), f_1\rangle),\\ \widehat{\delta}_2(\langle\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y))), f_2\rangle)\\ \in [\![(|\tau_1| \times \beta \Rightarrow |\tau_2|) \times \beta]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

Where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow [\tau'/\alpha](\tau_1 \to \tau_2)$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow [\tau'/\alpha](\tau_1 \to \tau_2)$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \texttt{R}'$. We already know that $(f_1, f_2) \in [\![\beta]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''} = \texttt{R}' = [\![[\tau'/\alpha](\tau_1 \to \tau_2)]\!]$ by assumption, so by definition we just need to show that

$$(\widehat{\delta}_1([\tau'/\alpha](\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1}(\pi_1 y))))),\\ \widehat{\delta}_2(\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y)))))\\ \in [\![|\tau_1| \times \beta \Rightarrow |\tau_2|]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

To do this we assume that $(v_1, v_2) \in [\![|\tau_1| \times \beta]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$ and show that

$$(\widehat{\delta}_1([\tau'/\alpha](\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1}(\pi_1 y)))))\widehat{\ }v_1,\\ \widehat{\delta}_2(\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y))))\widehat{\ }v_2)\\ \in [\![|\tau_2|]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

So by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(([\tau'/\alpha]\texttt{over}_{\tau_2})((\pi_2 v_1)\ (([\tau'/\alpha]\texttt{back}_{\tau_1})(\pi_1 v_1)))),\\ \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 v_2)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))))) \in [\![|\tau_2|]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

However by induction we know that

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\tau_2}), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau_2})) \in [\![[\tau'/\alpha]\tau_2 \to |\tau_2|]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

So by Lemma 8.3 we just need to show that

$$(\widehat{\delta}_1((\pi_2 v_1)\ (([\tau'/\alpha]\texttt{back}_{\tau_1})(\pi_1 v_1))), \widehat{\delta}_2((\pi_2 v_2)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2)))) \in [\![[\tau'/\alpha]\tau_2]\!]^{\texttt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

By assumption we know that $(\pi_2 v_1, \pi_2 v_2) \in [\![\beta]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'} = [\![[\tau'/\alpha](\tau_1 \to \tau_2)]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$, so again by Lemma 8.3 this is the same as showing that

$$(\widehat{\delta}_1(([\tau'/\alpha]\mathtt{back}_{\tau_1})(\pi_1 v_1)), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))) \in [\![[\tau'/\alpha]\tau_1]\!]^{\mathtt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

We also know by induction that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\tau_1}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_1})) \in [\![|\tau_1| \to [\tau'/\alpha]\tau_1]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

So by Lemma 8.3 all we have to do to show the above is show that

$$(\widehat{\delta}_1(\pi_1 v_1), \widehat{\delta}_2(\pi_1 v_2)) \in [\![|\tau_1|]\!]^{\mathtt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

However this follows from our assumption that $(v_1, v_2) \in [\![|\tau_1| \times \beta]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$, and so the result follows.

Case for $\tau = \forall\beta.\tau''$

In this case we want to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\forall\beta.\tau''}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha](\forall\beta.\tau'')})) \in [\![|\forall\beta.\tau''| \to [\tau'/\alpha](\forall\beta.\tau'')]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

Then we assume that $(v_1, v_2) \in [\![|\forall\beta.\tau''|]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$, and so by definition we just need to show

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_{\forall\beta.\tau''})\ v_1, \widehat{\delta}_2(\mathtt{back}_{\forall\beta.([\tau'/\alpha]\tau'')})\ v_2) \in [\![\forall\beta.([\tau'/\alpha]\tau'')]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

By the definition of back, this is equivalent to

$$\begin{aligned}&(\widehat{\delta}_1([\tau'/\alpha](\lambda x : |\forall\beta.\tau''|.\Lambda\beta.\mathtt{back}_{\tau''}(x[\beta]))\ v_1),\\ &\ \widehat{\delta}_2((\lambda x : |\forall\beta.\tau''|.\Lambda\beta.\mathtt{back}_{[\tau'/\alpha]\tau''}(x[\beta]))\ v_2)) \in [\![\forall\beta.([\tau'/\alpha]\tau'')]\!]^{\mathtt{E}}_{\eta':\delta_1' \leftrightarrow \delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\Lambda\beta.([\tau'/\alpha]\mathtt{back}_{\tau''})(v_1[\beta])),\\ &\ \widehat{\delta}_2(\Lambda\beta.\mathtt{back}_{[\tau'/\alpha]\tau''}(v_2[\beta]))) \in [\![\forall\beta.([\tau'/\alpha]\tau'')]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}\end{aligned}$$

Then by definition, we just need to show that for all τ_1 $\mathtt{type}$, τ_2 $\mathtt{type}$, and $\mathtt{R}' \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$ s.t. $\mathtt{R}' = \mathtt{R}'^{\mathtt{ST}}$,

$$\begin{aligned}&(\widehat{\delta}_1(\Lambda\beta.([\tau'/\alpha]\mathtt{back}_{\tau''})(v_1[\beta]))[\tau_1],\\ &\ \widehat{\delta}_2(\Lambda\beta.\mathtt{back}_{[\tau'/\alpha]\tau''}(v_2[\beta]))[\tau_2]) \in [\![([\tau'/\alpha]\tau'')]\!]^{\mathtt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}\end{aligned}$$

Where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \mathtt{R}'$. So by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(([\tau'/\alpha]\mathtt{back}_{\tau''})(v_1[\tau_1])), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau''}(v_2[\tau_2]))) \in [\![([\tau'/\alpha]\tau'')]\!]^{\mathtt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

However, we get by induction that

$$(\widehat{\delta}_1'([\tau'/\alpha]\mathtt{back}_{\tau''}), \widehat{\delta}_2'(\mathtt{back}_{[\tau'/\alpha]\tau''})) \in [\![|\tau''| \to [\tau'/\alpha]\tau'']\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

And by our assumption and the definition of the type relation we know that

$$(v_1[\tau_1], v_2[\tau_2]) \in [\![|\tau''|]\!]^{\mathtt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

So with the above, the desired result follows by Lemma 8.3 for functions.

Now we want to show the same for over, which is that

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\forall\beta.\tau''}), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha](\forall\beta.\tau'')})) \in [\![[\tau'/\alpha](\forall\beta.\tau'') \to |\forall\beta.\tau''|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then we assume that $(v_1, v_2) \in [\![[\tau'/\alpha](\forall\beta.\tau'')]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and so by definition we just need to show

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\forall\beta.\tau''})\ v_1, \widehat{\delta}_2(\texttt{over}_{\forall\beta.([\tau'/\alpha]\tau'')})\ v_2) \in [\![\forall\beta.|\tau''|]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of over, this is equivalent to

$$\begin{aligned}&(\widehat{\delta}_1([\tau'/\alpha](\lambda x : (\forall\beta.\tau'').\Lambda\beta.\texttt{over}_{\tau''}(x[\beta]))\ v_1),\\ &\ \widehat{\delta}_2((\lambda x : (\forall\beta.\tau'').\Lambda\beta.\texttt{over}_{[\tau'/\alpha]\tau''}(x[\beta]))\ v_2)) \in [\![\forall\beta.|\tau''|]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(\Lambda\beta.([\tau'/\alpha]\texttt{over}_{\tau''})(v_1[\beta])), \widehat{\delta}_2(\Lambda\beta.\texttt{over}_{[\tau'/\alpha]\tau''}(v_2[\beta]))) \in [\![\forall\beta.|\tau''|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then by definition, we just need to show that for all τ_1 type, τ_2 type, and $\texttt{R}' \subseteq \texttt{Val}(\tau_1) \times \texttt{Val}(\tau_2)$ s.t. $\texttt{R}' = \texttt{R}'^{\texttt{ST}}$,

$$(\widehat{\delta}_1(\Lambda\beta.([\tau'/\alpha]\texttt{over}_{\tau''})(v_1[\beta])[\tau_1]), \widehat{\delta}_2(\Lambda\beta.\texttt{over}_{[\tau'/\alpha]\tau''}(v_2[\beta]))[\tau_2]) \in [\![|\tau''|]\!]^{\texttt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

Where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \texttt{R}'$. So by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(([\tau'/\alpha]\texttt{over}_{\tau''})(v_1[\tau_1])), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau''}(v_2[\tau_2]))) \in [\![|\tau''|]\!]^{\texttt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

However, we get by induction that

$$(\widehat{\delta}_1'([\tau'/\alpha]\texttt{over}_{\tau''}), \widehat{\delta}_2'(\texttt{over}_{[\tau'/\alpha]\tau''})) \in [\![[\tau'/\alpha]\tau'' \to |\tau''|]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

And by our assumption and the definition of the type relation we know that

$$(v_1[\tau_1], v_2[\tau_2]) \in [\![[\tau'/\alpha]\tau'']\!]^{\texttt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

So with the above, the desired result follows by Lemma 8.3 for functions.

Case for $\tau = \exists\beta.\tau''$
In this case we want to show that

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{back}_{\exists\beta.\tau''}), \widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha](\exists\beta.\tau'')})) \in [\![|\exists\beta.\tau''| \to [\tau'/\alpha](\exists\beta.\tau'')]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then we assume that $(v_1, v_2) \in [\![|\exists\beta.\tau''|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and so by definition we just need to show

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{back}_{\exists\beta.\tau''})\ v_1, \widehat{\delta}_2(\texttt{back}_{\exists\beta.([\tau'/\alpha]\tau'')})\ v_2) \in [\![\exists\beta.([\tau'/\alpha]\tau'')]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of back, this is equivalent to

$$\begin{aligned}&(\widehat{\delta}_1([\tau'/\alpha](\lambda x : |\exists\beta.\tau''|.\,\texttt{unpack}[\beta, y] = x\,\texttt{in}(\texttt{pack}[\beta, \texttt{back}_{\tau''}(y)]\,\texttt{as}\,\exists\beta.\tau''))\ v_1),\\ &\ \widehat{\delta}_2((\lambda x : |\exists\beta.[\tau'/\alpha]\tau''|.\,\texttt{unpack}[\beta, y] = x\,\texttt{in}(\texttt{pack}[\beta, \texttt{back}_{[\tau'/\alpha]\tau''}(y)]\,\texttt{as}\,\exists\beta.[\tau'/\alpha]\tau''))\ v_2))\\ &\ \in [\![\exists\beta.([\tau'/\alpha]\tau'')]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\texttt{unpack}[\beta, y] = v_1 \;\texttt{in}(\texttt{pack}[\beta, ([\tau'/\alpha]\texttt{back}_{\tau''})(y)] \;\texttt{as}\; \exists\beta.[\tau'/\alpha]\tau'')),\\ &\widehat{\delta}_2(\texttt{unpack}[\beta, y] = v_2 \;\texttt{in}(\texttt{pack}[\beta, \texttt{back}_{[\tau'/\alpha]\tau''}(y)] \;\texttt{as}\; \exists\beta.[\tau'/\alpha]\tau'')))\\ &\in [\![\exists\beta.([\tau'/\alpha]\tau'')]\!]^{\texttt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}\end{aligned}$$

Since $(v_1, v_2) \in [\![\exists\beta.|\tau''|]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$ by assumption, we know by definition that there exist $\tau_1\,\texttt{type}$, $\tau_2\,\texttt{type}$, $e_1\,\texttt{val}$, $e_2\,\texttt{val}$ s.t. $v_1 = \texttt{pack}[\tau_1, e_1]\;\texttt{as}\;\exists\beta.[\tau'/\alpha]|\tau''|$ and $v_2 = \texttt{pack}[\tau_2, e_2]\;\texttt{as}\;\exists\beta.|[\tau'/\alpha]\tau''|$, as well as a relation $\texttt{R}'$ such that $(e_1, e_2) \in [\![|\tau''|]\!]_{\eta'':\delta''_1\leftrightarrow\delta''_2}$, where $\delta''_1 = \delta'_1 \otimes \beta \hookrightarrow \tau_1$, $\delta''_2 = \delta'_2 \otimes \beta \hookrightarrow \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \texttt{R}'$. Thus we have that equivalently,

$$\begin{aligned}&(\widehat{\delta}_1(\texttt{unpack}[\beta, y] = \texttt{pack}[\tau_1, e_1]\;\texttt{as}\;\exists\beta.[\tau'/\alpha]|\tau''| \;\texttt{in}(\texttt{pack}[\beta, ([\tau'/\alpha]\texttt{back}_{\tau''})(y)] \;\texttt{as}\; \exists\beta.[\tau'/\alpha]\tau'')),\\ &\widehat{\delta}_2(\texttt{unpack}[\beta, y] = \texttt{pack}[\tau_2, e_2]\;\texttt{as}\;\exists\beta.|[\tau'/\alpha]\tau''| \;\texttt{in}(\texttt{pack}[\beta, \texttt{back}_{[\tau'/\alpha]\tau''}(y)] \;\texttt{as}\; \exists\beta.[\tau'/\alpha]\tau'')))\\ &\in [\![\exists\beta.([\tau'/\alpha]\tau'')]\!]^{\texttt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\texttt{pack}[\tau_1, ([\tau_1/\beta][\tau'/\alpha]\texttt{back}_{\tau''})(e_1)] \;\texttt{as}\; \exists\beta.[\tau'/\alpha]\tau''),\\ &\widehat{\delta}_2(\texttt{pack}[\tau_2, ([\tau_2/\beta]\texttt{back}_{[\tau'/\alpha]\tau''})(e_2)] \;\texttt{as}\; \exists\beta.[\tau'/\alpha]\tau''))\\ &\in [\![\exists\beta.([\tau'/\alpha]\tau'')]\!]^{\texttt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}\end{aligned}$$

We can then use the types and values from v_1 and v_2 as the ones that exist to make this hold by using Lemma 8.3 for existentials, and use the relation $\texttt{R}'$ so we just have to show that

$$(\widehat{\delta}_1(([\tau_1/\beta][\tau'/\alpha]\texttt{back}_{\tau''})(e_1)), \widehat{\delta}_2(([\tau_2/\beta]\texttt{back}_{[\tau'/\alpha]\tau''})(e_2))) \in [\![[\tau'/\alpha]\tau'']\!]^{\texttt{E}}_{\eta'':\delta''_1\leftrightarrow\delta''_2}$$

However by induction we get that

$$(\widehat{\delta}_1([\tau_1/\beta][\tau'/\alpha]\texttt{back}_{\tau''}), \widehat{\delta}_2([\tau_2/\beta]\texttt{back}_{[\tau'/\alpha]\tau''})) \in [\![|\tau''| \to [\tau'/\alpha]\tau'']\!]_{\eta'':\delta''_1\leftrightarrow\delta''_2}$$

And we already know by our assumption as shown earlier that $(e_1, e_2) \in [\![|\tau''|]\!]_{\eta'':\delta''_1\leftrightarrow\delta''_2}$, so combining this with what we get by induction, the desired result follows.

Now we want to show the same for over, which is that

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\exists\beta.\tau''}), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha](\exists\beta.\tau'')})) \in [\![[\tau'/\alpha](\exists\beta.\tau'') \to |\exists\beta.\tau''|]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

Then we assume that $(v_1, v_2) \in [\![[\tau'/\alpha](\exists\beta.\tau'')]\!]_{\eta':\delta'_1\leftrightarrow\delta'_2}$, and so by definition we just need to show

$$(\widehat{\delta}_1([\tau'/\alpha]\texttt{over}_{\exists\beta.\tau''})\; v_1, \widehat{\delta}_2(\texttt{over}_{\exists\beta.([\tau'/\alpha]\tau'')})\; v_2) \in [\![|\exists\beta.\tau''|]\!]^{\texttt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

By the definition of over, this is equivalent to

$$\begin{aligned}&(\widehat{\delta}_1([\tau'/\alpha](\lambda x : \exists\beta.\tau''.\,\texttt{unpack}[\beta, y] = x \;\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{\tau''}(y)] \;\texttt{as}\; |\exists\beta.\tau''|))\; v_1),\\ &\widehat{\delta}_2((\lambda x : \exists\beta.[\tau'/\alpha]\tau''.\,\texttt{unpack}[\beta, y] = x \;\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{[\tau'/\alpha]\tau''}(y)] \;\texttt{as}\; |\exists\beta.[\tau'/\alpha]\tau''|))\; v_2))\\ &\in [\![|\exists\beta.\tau''|]\!]^{\texttt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1([\tau'/\alpha]\,\texttt{unpack}[\beta, y] = v_1 \;\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{\tau''}(y)] \;\texttt{as}\; |\exists\beta.\tau''|)),\\ &\widehat{\delta}_2(\texttt{unpack}[\beta, y] = v_2 \;\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{[\tau'/\alpha]\tau''}(y)] \;\texttt{as}\; |\exists\beta.[\tau'/\alpha]\tau''|)))\\ &\in [\![|\exists\beta.\tau''|]\!]^{\texttt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}\end{aligned}$$

Since $(v_1, v_2) \in [\![\exists\beta.[\tau'/\alpha]\tau'']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$ by assumption, we know by definition that there exist τ_1 $\texttt{type}$, τ_2 $\texttt{type}$, e_1 $\texttt{val}$, e_2 $\texttt{val}$ s.t. $v_1 = \texttt{pack}[\tau_1, e_1]\,\texttt{as}\,\exists\beta.[\tau'/\alpha]\tau''$ and $v_2 = \texttt{pack}[\tau_2, e_2]\,\texttt{as}\,\exists\beta.[\tau'/\alpha]\tau''$, as well as a relation $\texttt{R}'$ such that $(e_1, e_2) \in [\![[\tau'/\alpha]\tau'']\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$, where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \texttt{R}'$. Thus we have that equivalently,

$$
\begin{aligned}
&(\widehat{\delta}_1([\tau'/\alpha]\,\texttt{unpack}[\beta, y] = \texttt{pack}[\tau_1, e_1]\,\texttt{as}\,\exists\beta.[\tau'/\alpha]\tau''\,\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{\tau''}(y)]\,\texttt{as}\,|\exists\beta.\tau''|)),\\
&\;\widehat{\delta}_2(\texttt{unpack}[\beta, y] = \texttt{pack}[\tau_2, e_2]\,\texttt{as}\,\exists\beta.[\tau'/\alpha]\tau''\,\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{[\tau'/\alpha]\tau''}(y)]\,\texttt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|)))\\
&\;\in [\![|\exists\beta.\tau''|]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}
\end{aligned}
$$

Then by Lemma 7.5, this is equivalent to showing

$$
\begin{aligned}
&(\widehat{\delta}_1(\texttt{pack}[\tau_1, ([\tau_1/\beta][\tau'/\alpha]\texttt{over}_{\tau''})(e_1)]\,\texttt{as}[\tau'/\alpha]|\exists\beta.\tau''|),\\
&\;\widehat{\delta}_2(\texttt{pack}[\tau_2, ([\tau_2/\beta]\texttt{over}_{[\tau'/\alpha]\tau''})(e_2)]\,\texttt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|))\\
&\;\in [\![\exists\beta.|\tau''|]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}
\end{aligned}
$$

We can then use the types and values from v_1 and v_2 as the ones that exist to make this hold by using Lemma 8.3 for existentials, and use the relation $\texttt{R}'$ so we just have to show that

$$
(\widehat{\delta}_1(([\tau_1/\beta][\tau'/\alpha]\texttt{over}_{\tau''})(e_1)), \widehat{\delta}_2(([\tau_2/\beta]\texttt{over}_{[\tau'/\alpha]\tau''})(e_2))) \in [\![|\tau''|]\!]^{\texttt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}
$$

However by induction we get that

$$
(\widehat{\delta}_1([\tau_1/\beta][\tau'/\alpha]\texttt{over}_{\tau''}), \widehat{\delta}_2([\tau_2/\beta]\texttt{over}_{[\tau'/\alpha]\tau''})) \in [\![[\tau'/\alpha]\tau'' \to |\tau''|]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}
$$

And we already know by our assumption as shown earlier that $(e_1, e_2) \in [\![[\tau'/\alpha]\tau'']\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$, so combining this with what we get by induction, the desired result follows.

□

11.4 The Over Relation

Lemma 11.4. If $\Delta, \alpha \vdash_S \tau\,\texttt{type}$ and $\Delta \vdash_S \tau'\,\texttt{type}$, then for any $\delta_1 : \Delta$, $\delta_2 : \Delta$, and $\eta : \delta_1 \leftrightarrow \delta_2$, there exists an ST-closed relation $\texttt{R} \in \texttt{Val}(\widehat{\delta}_1(|\tau'|)) \times \texttt{Val}(\widehat{\delta}_2(\tau'))$ defined as

$$\texttt{R} = \{(v_1, v_2) \,|\, (v_1, \widehat{\delta}_2(\texttt{over}_{\tau'})(v_2)) \in [\![|\tau'|]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\}$$

such that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{back}_\tau), \widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha]\tau})) \in [\![|[\tau'/\alpha]\tau| \to \tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{over}_\tau), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau})) \in [\![\tau \to |[\tau'/\alpha]\tau|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Where $\eta' = \eta \otimes \alpha \hookrightarrow \texttt{R}$, $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \widehat{\delta}_1(|\tau'|)$, $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \widehat{\delta}_2(\tau')$, We know that $\texttt{R}$ is ST-closed because it is defined in terms of $[\![|\tau'|]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$, which is itself ST-closed.

Proof. The proof is done by induction on the structure τ, and follows in much the same manner as Lemma 11.3.

Case for $\tau = \texttt{unit}$
In this case we want to show that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{back}_{\texttt{unit}}), \widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha]\texttt{unit}})) \in [\![|[\tau'/\alpha]\texttt{unit}| \to \texttt{unit}]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This is equivalent to

$$(\texttt{back}_{\texttt{unit}}, \texttt{back}_{\texttt{unit}}) \in [\![\texttt{unit} \to \texttt{unit}]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This follows immediately from Reflexivity. The case for `over` works exactly the same.

Case for $\tau = \texttt{int}$
In this case we want to show that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{back}_{\texttt{int}}), \widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha]\texttt{int}})) \in [\![|[\tau'/\alpha]\texttt{int}| \to \texttt{int}]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This is equivalent to

$$(\texttt{back}_{\texttt{int}}, \texttt{back}_{\texttt{int}}) \in [\![\texttt{int} \to \texttt{int}]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This follows immediately from Reflexivity. The case for `over` works exactly the same.

Case for $\tau = \alpha$
In this case we want to show that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{over}_\alpha), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\alpha})) \in [\![\alpha \to |[\tau'/\alpha]\alpha|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This is equivalent to

$$(\widehat{\delta}_1([|\tau'|/\alpha](\lambda x : \alpha.x)), \widehat{\delta}_2(\texttt{over}_{\tau'})) \in [\![\alpha \to |\tau'|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

And similarly equivalent to

$$(\lambda x : \widehat{\delta}_1(|\tau'|).x, \widehat{\delta}_2(\texttt{over}_{\tau'})) \in [\![\alpha \to |\tau'|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

To show this we assume that $(v_1, v_2) \in [\![\alpha]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, which implies that $(v_1, v_2) \in \texttt{R}$, and show that

$$((\lambda x : \widehat{\delta}_1(|\tau'|).x)\ v_1, \widehat{\delta}_2(\texttt{over}_{\tau'})\ v_2) \in [\![|\tau'|]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

However, since $|\tau'|$ doesn't reference α, we can equivalently show that

$$(v_1, \widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2) \in [\![|\tau'|]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

This follows immediately from our definition of $\mathtt{R}$.

Now we need to show the other half, that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_\alpha), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\alpha})) \in [\![|[\tau'/\alpha]\alpha| \to |\alpha|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This is equivalent to

$$(\widehat{\delta}_1([|\tau'|/\alpha](\lambda x : \alpha.x)), \widehat{\delta}_2(\mathtt{back}_{\tau'})) \in [\![|\tau'| \to \alpha]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

And similarly equivalent to

$$(\lambda x : \widehat{\delta}_1(|\tau'|).x, \widehat{\delta}_2(\mathtt{back}_{\tau'})) \in [\![|\tau'| \to \alpha]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

To show this we assume that $(v_1, v_2) \in [\![|\tau'|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and show that

$$((\lambda x : \widehat{\delta}_1(|\tau'|).x)\ v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2) \in [\![\alpha]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

This is equivalent to showing that

$$(v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2) \in [\![\alpha]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Which is the same as showing that $(v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2) \in \mathtt{R}^{\mathtt{E}}$. We know by Lemma 11.1 that

$$v_2 \cong_{\widehat{\delta}_2(\tau')} \widehat{\delta}_2(\mathtt{over}_{\tau'})(\widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2)$$

So by Lemma 9.3, we know from our assumption that

$$(v_1, \widehat{\delta}_2(\mathtt{over}_{\tau'})(\widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2)) \in [\![|\tau'|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Since $v_1 \downarrow$, this implies that $\widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2 \downarrow$, so we have that $\widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2 \mapsto^* v_2'$ for some value v_2'. Then we know by Lemma 7.5 that

$$(v_1, \widehat{\delta}_2(\mathtt{over}_{\tau'})\ v_2') \in [\![|\tau'|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

But by definition this implies that $(v_1, v_2') \in \mathtt{R}$, and therefore that $(v_1, \widehat{\delta}_2(\mathtt{back}_{\tau'})\ v_2) \in \mathtt{R}^{\mathtt{E}}$.

Case for $\tau = \tau_1 \times \tau_2$

In this case we want to show that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\tau_1\times\tau_2}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha](\tau_1\times\tau_2)})) \in [\![\tau_1 \times \tau_2 \to |[\tau'/\alpha](\tau_1 \times \tau_2)|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

To show this we assume that $(v_1, v_2) \in [\![\tau_1 \times \tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and show

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\tau_1\times\tau_2})\ v_1, \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha](\tau_1\times\tau_2)})\ v_2) \in [\![|[\tau'/\alpha](\tau_1 \times \tau_2)|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of over, this is the same as showing

$$\begin{aligned}&(\widehat{\delta}_1(([|\tau'|/\alpha](\lambda x : \tau_1 \times \tau_2.\langle \mathtt{over}_{\tau_1}(\pi_1 x), \mathtt{over}_{\tau_2}(\pi_2 x)\rangle))(v_1)),\\ &\widehat{\delta}_2(\lambda x : [\tau'/\alpha](\tau_1 \times \tau_2).\langle \mathtt{over}_{[\tau'/\alpha]\tau_1}(\pi_1 x), \mathtt{over}_{[\tau'/\alpha]\tau_2}(\pi_2 x)\rangle(v_2))) \in [\![|[\tau'/\alpha](\tau_1 \times \tau_2)|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

By Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(\langle([|\tau'|/\alpha]\mathtt{over}_{\tau_1})(\pi_1 v_1), ([|\tau'|/\alpha]\mathtt{over}_{\tau_2})(\pi_2 v_1)\rangle),$$
$$\widehat{\delta}_2(\langle \mathtt{over}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2), \mathtt{over}_{[\tau'/\alpha]\tau_2}(\pi_2 v_2)\rangle)) \in [\![|[\tau'/\alpha]\tau_1| \times |[\tau'/\alpha]\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then using Lemma 8.3 for pairs we just need to show that

$$(\widehat{\delta}_1(([|\tau'|/\alpha]\mathtt{over}_{\tau_1})(\pi_1 v_1)), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))) \in [\![|[\tau'/\alpha]\tau_1|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1(([|\tau'|/\alpha]\mathtt{over}_{\tau_2})(\pi_2 v_1)), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_2}(\pi_2 v_2))) \in [\![|[\tau'/\alpha]\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

However by induction we get that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\tau_1}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_1})) \in [\![\tau_1 \to |[\tau'/\alpha]\tau_1|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\tau_2}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_2})) \in [\![\tau_2 \to |[\tau'/\alpha]\tau_2|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We also know by our assumption and the definition of the type relation that $(\pi_1 v_1, \pi_1 v_2) \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$ and $(\pi_2 v_1, \pi_2 v_2) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$, from which the desired result follows by Lemma 8.3 for functions.

Now we want to show this for `back` as well, so we want to show that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\tau_1\times\tau_2}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha](\tau_1\times\tau_2)})) \in [\![|[\tau'/\alpha](\tau_1 \times \tau_2)| \to \tau_1 \times \tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

To show this we assume that $(v_1, v_2) \in [\![|[\tau'/\alpha](\tau_1 \times \tau_2)|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and show

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\tau_1\times\tau_2})\ v_1, \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha](\tau_1\times\tau_2)})\ v_2) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of `back`, this is the same as showing

$$(\widehat{\delta}_1([|\tau'|/\alpha]\lambda x : |\tau_1 \times \tau_2|.\langle \mathtt{back}_{\tau_1}(\pi_1 x), \mathtt{back}_{\tau_2}(\pi_2 x)\rangle\ v_1),$$
$$\widehat{\delta}_2(\lambda x : |[\tau'/\alpha](\tau_1 \times \tau_2)|.\langle \mathtt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 x), \mathtt{back}_{[\tau'/\alpha]\tau_2}(\pi_2 x)\rangle\ v_2)) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(\langle([|\tau'|/\alpha]\mathtt{back}_{\tau_1})(\pi_1 v_1), ([|\tau'|/\alpha]\mathtt{back}_{\tau_2})(\pi_2 v_1)\rangle),$$
$$\widehat{\delta}_2(\langle \mathtt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2), \mathtt{back}_{[\tau'/\alpha]\tau_2}(\pi_2 v_2)\rangle)) \in [\![\tau_1 \times \tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then using Lemma 8.3 for pairs we just need to show that

$$(\widehat{\delta}_1(([|\tau'|/\alpha]\mathtt{back}_{\tau_1})(\pi_1 v_1)), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))) \in [\![\tau_1]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1(([|\tau'|/\alpha]\mathtt{back}_{\tau_2})(\pi_2 v_1)), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_2}(\pi_2 v_2))) \in [\![\tau_2]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

However by induction we get that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\tau_1}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_1})) \in [\![|[\tau'/\alpha]\tau_1| \to \tau_1]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\tau_2}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_2})) \in [\![|[\tau'/\alpha]\tau_2| \to \tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We also know by our assumption and the definition of the type relation that $(\pi_1 v_1, \pi_1 v_2) \in [\![|[\tau'/\alpha]\tau_1|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$ and $(\pi_2 v_1, \pi_2 v_2) \in [\![|[\tau'/\alpha]\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$, from which the desired result follows by Lemma 8.3 for functions.

Case for $\tau = \tau_1 \to \tau_2$
In this case we want to show that

$$(\widehat{\delta_1}([|\tau'|/\alpha]\texttt{back}_{\tau_1\to\tau_2}), \widehat{\delta_2}(\texttt{back}_{[\tau'/\alpha](\tau_1\to\tau_2)})) \in [\![|[\tau'/\alpha](\tau_1 \to \tau_2)| \to \tau_1 \to \tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We assume that

$$(f_1, f_2) \in [\![|[\tau'/\alpha](\tau_1 \to \tau_2)|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'} = [\![\exists\beta.(|[\tau'/\alpha]\tau_1| \times \beta \Rightarrow |[\tau'/\alpha]\tau_2|) \times \beta]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

So by definition we just need to show that

$$(\widehat{\delta_1}([|\tau'|/\alpha]\texttt{back}_{\tau_1\to\tau_2})\ f_1, \widehat{\delta_2}(\texttt{back}_{[\tau'/\alpha]\tau_1\to[\tau'/\alpha]\tau_2})\ f_2) \in [\![\tau_1 \to \tau_2]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of `back` this is the same as showing

$$\begin{aligned}&(\widehat{\delta_1}(([|\tau'|/\alpha](\lambda f : |\tau_1 \to \tau_2|.\lambda y : \tau_1.\,\texttt{unpack}[\beta, g] = f\,\texttt{in}\,\texttt{back}_{\tau_2}((\pi_1 g)\hat{}\langle\texttt{over}_{\tau_1} y, \pi_2 g\rangle)))\ f_1),\\ &\widehat{\delta_2}((\lambda f : |[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2|.\lambda y : [\tau'/\alpha]\tau_1.\,\texttt{unpack}[\beta, g] = f\ \texttt{in}\\ &\quad \texttt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle\texttt{over}_{[\tau'/\alpha]\tau_1} y, \pi_2 g\rangle))\ f_2)) \in [\![\tau_1 \to \tau_2]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta_1}((\lambda y : [|\tau'|/\alpha]\tau_1.\,\texttt{unpack}[\beta, g] = f_1\ \texttt{in}([|\tau'|/\alpha]\texttt{back}_{\tau_2})((\pi_1 g)\hat{}\langle([|\tau'|/\alpha]\texttt{over}_{\tau_1})y, \pi_2 g\rangle))),\\ &\widehat{\delta_2}(\lambda y : [\tau'/\alpha]\tau_1.\,\texttt{unpack}[\beta, g] = f_2\,\texttt{in}\,\texttt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle\texttt{over}_{[\tau'/\alpha]\tau_1} y, \pi_2 g\rangle)))\\ &\quad\in [\![\tau_1 \to \tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

To show this we assume that $(v_1, v_2) \in [\![\tau_1]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$ and show that

$$\begin{aligned}&(\widehat{\delta_1}(\lambda y : [|\tau'|/\alpha]\tau_1.\,\texttt{unpack}[\beta, g] = f_1\ \texttt{in}([|\tau'|/\alpha]\texttt{back}_{\tau_2})((\pi_1 g)\hat{}\langle([|\tau'|/\alpha]\texttt{over}_{\tau_1})y, \pi_2 g\rangle))\ v_1,\\ &\widehat{\delta_2}(\lambda y : [\tau'/\alpha]\tau_1.\,\texttt{unpack}[\beta, g] = f_2\,\texttt{in}\,\texttt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle\texttt{over}_{[\tau'/\alpha]\tau_1} y, \pi_2 g\rangle))\ v_2)\\ &\quad\in [\![\tau_2]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta_1}(\texttt{unpack}[\beta, g] = f_1\ \texttt{in}([|\tau'|/\alpha]\texttt{back}_{\tau_2})((\pi_1 g)\hat{}\langle([|\tau'|/\alpha]\texttt{over}_{\tau_1})v_1, \pi_2 g\rangle)),\\ &\widehat{\delta_2}(\texttt{unpack}[\beta, g] = f_2\,\texttt{in}\,\texttt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle\texttt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g\rangle)))\\ &\quad\in [\![\tau_2]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Since f_1 is a value, we know by rule $Vpack$ that it is of the form $\texttt{pack}[\tau_1', g_1]\,\texttt{as}\,|[\tau'/\alpha](\tau_1 \to \tau_2)|$ for appropriate type τ_1' and value g_1. Similarly we can write f_2 as $\texttt{pack}[\tau_2', g_2]\,\texttt{as}\,|[\tau'/\alpha](\tau_1 \to \tau_2)|$. Thus we can show that

$$\begin{aligned}&(\widehat{\delta_1}(\texttt{unpack}[\beta, g] = \texttt{pack}[\tau_1', g_1]\,\texttt{as}\,|[\tau'/\alpha](\tau_1 \to \tau_2)|\,\texttt{in}\\ &\quad([|\tau'|/\alpha]\texttt{back}_{\tau_2})((\pi_1 g)\hat{}\langle([|\tau'|/\alpha]\texttt{over}_{\tau_1})v_1, \pi_2 g\rangle)),\\ &\widehat{\delta_2}(\texttt{unpack}[\beta, g] = \texttt{pack}[\tau_2', g_2]\,\texttt{as}\,|[\tau'/\alpha](\tau_1 \to \tau_2)|\,\texttt{in}\,\texttt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g)\hat{}\langle\texttt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g\rangle)))\\ &\quad\in [\![\tau_2]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Thus by Lemma 7.5, we can just show that

$$\begin{aligned}&(\widehat{\delta_1}(([|\tau'|/\alpha]\texttt{back}_{\tau_2})((\pi_1 g_1)\hat{}\langle([|\tau'|/\alpha]\texttt{over}_{\tau_1})v_1, \pi_2 g_1\rangle)),\\ &\widehat{\delta_2}(\texttt{back}_{[\tau'/\alpha]\tau_2}((\pi_1 g_2)\hat{}\langle\texttt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g_2\rangle))) \in [\![\tau_2]\!]^{\texttt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}\end{aligned}$$

However since we know by induction that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\tau_2}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau_2})) \in [\![|[\tau'/\alpha]\tau_2| \to \tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

So by Lemma 8.3 we just have to show that

$$\widehat{\delta}_1((\pi_1 g_1)\widehat{\ }\langle([|\tau'|/\alpha]\mathtt{over}_{\tau_1})v_1, \pi_2 g_1\rangle), \widehat{\delta}_2((\pi_1 g_2)\widehat{\ }\langle\mathtt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g_2\rangle)) \in [\![|[\tau'/\alpha]\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Now by our assumption that $(f_1, f_2) \in [\![|[\tau'/\alpha](\tau_1 \to \tau_2)|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, we know that there exists a relation $\mathtt{R}' \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$ such that

$$(g_1, g_2) \in [\![(|[\tau'/\alpha]\tau_1| \times \beta \Rightarrow |[\tau'/\alpha]\tau_2|) \times \beta]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

Where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1'$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2'$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \mathtt{R}'$. This tells us that by definition,

$$(\pi_1 g_1, \pi_1 g_2) \in [\![|[\tau'/\alpha]\tau_1| \times \beta \Rightarrow |[\tau'/\alpha]\tau_2|]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

$$(\pi_2 g_1, \pi_2 g_2) \in [\![\beta]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

We also know by induction that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\tau_1}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_1})) \in [\![\tau_1 \to |[\tau'/\alpha]\tau_1|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Which by definition tells us that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\tau_1})\ v_1, \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_1})\ v_2) \in [\![|[\tau'/\alpha]\tau_1|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Which then by Lemma 8.3 for pairs we know that

$$(\widehat{\delta}_1(\langle[|\tau'|/\alpha]\mathtt{over}_{\tau_1})\ v_1, \pi_2 g_1\rangle, \widehat{\delta}_2(\langle\mathtt{over}_{[\tau'/\alpha]\tau_1})\ v_2, \pi_2 g_1\rangle) \in [\![|[\tau'/\alpha]\tau_1| \times \beta]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

Then again by Lemma 8.3 for functions we know that

$$\widehat{\delta}_1((\pi_1 g_1)\widehat{\ }\langle([|\tau'|/\alpha]\mathtt{over}_{\tau_1})v_1, \pi_2 g_1\rangle), \widehat{\delta}_2((\pi_1 g_2)\widehat{\ }\langle\mathtt{over}_{[\tau'/\alpha]\tau_1} v_2, \pi_2 g_2\rangle)) \in [\![|[\tau'/\alpha]\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Which no longer depends on β, so it is what we wanted to show.

Now we need to show the corresponding case for `over`, that is

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\tau_1\to\tau_2}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha](\tau_1\to\tau_2)})) \in [\![(\tau_1 \to \tau_2) \to |[\tau'/\alpha](\tau_1 \to \tau_2)|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We assume that

$$(f_1, f_2) \in [\![\tau_1 \to \tau_2]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

So by definition we just need to show that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\tau_1\to\tau_2})\ f_1, \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau_1\to[\tau'/\alpha]\tau_2})\ f_2) \in [\![|[\tau'/\alpha](\tau_1 \to \tau_2)|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of `over` this is the same as showing

$$\begin{aligned}
&(\widehat{\delta}_1(([|\tau'|/\alpha](\lambda f : \tau_1 \to \tau_2.\, \mathtt{pack}[\tau_1 \to \tau_2, \langle\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2). \\
&\quad \mathtt{over}_{\tau_2}((\pi_2 y)\ (\mathtt{back}_{\tau_1}(\pi_1 y))), f\rangle]\ \mathtt{as}\ |\tau_1 \to \tau_2|))\ f_1), \\
&\ \widehat{\delta}_2((\lambda f : [\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2.\, \mathtt{pack}[\tau_1 \to \tau_2, \langle\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2). \\
&\quad \mathtt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\mathtt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y))), f\rangle]\ \mathtt{as}\ |[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2|)\ f_2)) \\
&\in [\![|[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}
\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\texttt{pack}[[|\tau'|/\alpha](\tau_1 \to \tau_2), \langle\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2).\\ &\quad \texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1}(\pi_1 y))), f_1\rangle]\ \texttt{as}\ |[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2|),\\ &\widehat{\delta}_2(\texttt{pack}[\tau_1 \to \tau_2, \langle\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\\ &\quad \texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y))), f_2\rangle]\ \texttt{as}\ |[\tau'/\alpha]\tau_1 \to [\tau'/\alpha]\tau_2|))\\ &\in [\![\exists\beta.(|[\tau'/\alpha]\tau_1| \times \beta \Rightarrow |[\tau'/\alpha]\tau_2|) \times \beta]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}\end{aligned}$$

To show this, we pick $\texttt{R}' = [\![\tau_1 \to \tau_2]\!]$ and use the definition of the type relation to show that

$$\begin{aligned}&(\widehat{\delta}_1([|\tau'|/\alpha]\langle\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1}(\pi_1 y))), f_1\rangle),\\ &\widehat{\delta}_2(\langle\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y))), f_2\rangle)\\ &\in [\![(|[\tau'/\alpha]\tau_1| \times \beta \Rightarrow |[\tau'/\alpha]\tau_2|) \times \beta]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}\end{aligned}$$

Where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1 \to \tau_2$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_1 \to \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \texttt{R}'$. We already know that $(f_1, f_2) \in [\![\beta]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''} = \texttt{R}' = [\![\tau_1 \to \tau_2]\!]$ by assumption, so by definition we just need to show that

$$\begin{aligned}&(\widehat{\delta}_1([|\tau'|/\alpha](\widehat{\lambda} y : |\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{\tau_2}((\pi_2 y)\ (\texttt{back}_{\tau_1}(\pi_1 y))))),\\ &\widehat{\delta}_2(\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y)))))\\ &\in [\![|[\tau'/\alpha]\tau_1| \times \beta \Rightarrow |[\tau'/\alpha]\tau_2|]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}\end{aligned}$$

To do this we assume that $(v_1, v_2) \in [\![|[\tau'/\alpha]\tau_1| \times \beta]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$ and show that

$$\begin{aligned}&(\widehat{\delta}_1((\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).([|\tau'|/\alpha]\texttt{over}_{\tau_2})((\pi_2 y)\ (([|\tau'|/\alpha]\texttt{back}_{\tau_1})(\pi_1 y)))))\widehat{\ } v_1,\\ &\widehat{\delta}_2(\widehat{\lambda} y : |[\tau'/\alpha]\tau_1| \times (\tau_1 \to \tau_2).\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 y)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 y))))\widehat{\ } v_2)\\ &\in [\![|[\tau'/\alpha]\tau_2|]\!]^{\texttt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}\end{aligned}$$

So by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(([|\tau'|/\alpha]\texttt{over}_{\tau_2})((\pi_2 v_1)\ (([|\tau'|/\alpha]\texttt{back}_{\tau_1})(\pi_1 v_1)))),\\ &\widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau_2}((\pi_2 v_2)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))))) \in [\![|[\tau'/\alpha]\tau_2|]\!]^{\texttt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}\end{aligned}$$

However by induction we know that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{over}_{\tau_2}), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau_2})) \in [\![\tau_2 \to |[\tau'/\alpha]\tau_2|]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

So by Lemma 8.3 we just need to show that

$$\begin{aligned}&(\widehat{\delta}_1((\pi_2 v_1)\ (([|\tau'|/\alpha]\texttt{back}_{\tau_1})(\pi_1 v_1))), \widehat{\delta}_2((\pi_2 v_2)\ (\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))))\\ &\qquad \in [\![\tau_2]\!]^{\texttt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}\end{aligned}$$

By assumption we know that $(\pi_2 v_1, \pi_2 v_2) \in [\![\beta]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'} = [\![\tau_1 \to \tau_2]\!]_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$, so again by Lemma 8.3 this is the same as showing that

$$(\widehat{\delta}_1(([|\tau'|/\alpha]\texttt{back}_{\tau_1})(\pi_1 v_1)), \widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha]\tau_1}(\pi_1 v_2))) \in [\![\tau_1]\!]^{\texttt{E}}_{\eta'':\delta_1'' \leftrightarrow \delta_2''}$$

We also know by induction that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{back}_{\tau_1}), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha]\tau_1})) \in [\![|[\tau'/\alpha]\tau_1| \to \tau_1]\!]_{\eta':\delta_1' \leftrightarrow \delta_2'}$$

So by Lemma 8.3 all we have to do to show the above is show that

$$(\widehat{\delta}_1(\pi_1 v_1), \widehat{\delta}_2(\pi_1 v_2)) \in [\![|[\tau'/\alpha]\tau_1|]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

However this follows from our assumption that $(v_1, v_2) \in [\![|[\tau'/\alpha]\tau_1| \times \beta]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$, and so the result follows.

Case for $\tau = \forall\beta.\tau''$
In this case we want to show that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\forall\beta.\tau''}), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha](\forall\beta.\tau'')})) \in [\![\forall\beta.\tau'' \to |[\tau'/\alpha](\forall\beta.\tau'')|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then we assume that $(v_1, v_2) \in [\![\forall\beta.\tau'']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and so by definition we just need to show

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{over}_{\forall\beta.\tau''})\ v_1, \widehat{\delta}_2(\mathtt{over}_{\forall\beta.([\tau'/\alpha]\tau'')})\ v_2) \in [\![|\forall\beta.([\tau'/\alpha]\tau'')|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of `over`, this is equivalent to

$$\begin{aligned}&(\widehat{\delta}_1([|\tau'|/\alpha](\lambda x : (\forall\beta.\tau'').\Lambda\beta.\mathtt{over}_{\tau''}(x[\beta]))\ v_1),\\ &\widehat{\delta}_2((\lambda x : (\forall\beta.\tau'').\Lambda\beta.\mathtt{over}_{[\tau'/\alpha]\tau''}(x[\beta]))\ v_2)) \in [\![|\forall\beta.([\tau'/\alpha]\tau'')|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\Lambda\beta.([|\tau'|/\alpha]\mathtt{over}_{\tau''})(v_1[\beta])),\\ &\widehat{\delta}_2(\Lambda\beta.\mathtt{over}_{[\tau'/\alpha]\tau''}(v_2[\beta]))) \in [\![\forall\beta.|[\tau'/\alpha]\tau''|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by definition, we just need to show that for all $\tau_1\ \mathtt{type}$, $\tau_2\ \mathtt{type}$, and $\mathtt{R}' \subseteq \mathtt{Val}(\tau_1) \times \mathtt{Val}(\tau_2)$ s.t. $\mathtt{R}' = \mathtt{R}'^{\mathtt{ST}}$,

$$\begin{aligned}&(\widehat{\delta}_1(\Lambda\beta.([|\tau'|/\alpha]\mathtt{over}_{\tau''})(v_1[\beta]))[\tau_1],\\ &\widehat{\delta}_2(\Lambda\beta.\mathtt{over}_{[\tau'/\alpha]\tau''}(v_2[\beta]))[\tau_2]) \in [\![|[\tau'/\alpha]\tau''|]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}\end{aligned}$$

Where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \mathtt{R}'$. So by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(([|\tau'|/\alpha]\mathtt{over}_{\tau''})(v_1[\tau_1])), \widehat{\delta}_2(\mathtt{over}_{[\tau'/\alpha]\tau''}(v_2[\tau_2]))) \in [\![|[\tau'/\alpha]\tau''|]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

However, we get by induction that

$$(\widehat{\delta}_1'([|\tau'|/\alpha]\mathtt{over}_{\tau''}), \widehat{\delta}_2'(\mathtt{over}_{[\tau'/\alpha]\tau''})) \in [\![\tau'' \to |[\tau'/\alpha]\tau''|]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

And by our assumption and the definition of the type relation we know that

$$(v_1[\tau_1], v_2[\tau_2]) \in [\![\tau'']\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

So with the above, the desired result follows by Lemma 8.3 for functions.

Now we want to show the same for `back`, which is that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\forall\beta.\tau''}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha](\forall\beta.\tau'')})) \in [\![|[\tau'/\alpha](\forall\beta.\tau'')| \to \forall\beta.\tau'']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then we assume that $(v_1, v_2) \in [\![|[\tau'/\alpha](\forall\beta.\tau'')|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and so by definition we just need to show

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\forall\beta.\tau''})\ v_1, \widehat{\delta}_2(\mathtt{back}_{\forall\beta.([\tau'/\alpha]\tau'')})\ v_2) \in [\![\forall\beta.\tau'']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of back, this is equivalent to

$$(\widehat{\delta}_1([|\tau'|/\alpha](\lambda x : (\forall\beta.\tau'').\Lambda\beta.\texttt{back}_{\tau''}(x[\beta]))\ v_1),$$
$$\widehat{\delta}_2((\lambda x : (\forall\beta.\tau'').\Lambda\beta.\texttt{back}_{[\tau'/\alpha]\tau''}(x[\beta]))\ v_2)) \in [\![\forall\beta.\tau'']\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(\Lambda\beta.([|\tau'|/\alpha]\texttt{back}_{\tau''})(v_1[\beta])), \widehat{\delta}_2(\Lambda\beta.\texttt{back}_{[\tau'/\alpha]\tau''}(v_2[\beta]))) \in [\![\forall\beta.\tau'']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then by definition, we just need to show that for all τ_1 type, τ_2 type, and $\texttt{R}' \subseteq \texttt{Val}(\tau_1) \times \texttt{Val}(\tau_2)$ s.t. $\texttt{R}' = \texttt{R}'^{\texttt{ST}}$,

$$(\widehat{\delta}_1(\Lambda\beta.([|\tau'|/\alpha]\texttt{back}_{\tau''})(v_1[\beta])[\tau_1]), \widehat{\delta}_2(\Lambda\beta.\texttt{back}_{[\tau'/\alpha]\tau''}(v_2[\beta]))[\tau_2]) \in [\![\tau'']\!]^{\texttt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

Where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \texttt{R}'$. So by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(([|\tau'|/\alpha]\texttt{back}_{\tau''})(v_1[\tau_1])), \widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha]\tau''}(v_2[\tau_2]))) \in [\![\tau'']\!]^{\texttt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

However, we get by induction that

$$(\widehat{\delta_1'}([|\tau'|/\alpha]\texttt{back}_{\tau''}), \widehat{\delta_2'}(\texttt{back}_{[\tau'/\alpha]\tau''})) \in [\![|[\tau'/\alpha]\tau''| \to \tau'']\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

And by our assumption and the definition of the type relation we know that

$$(v_1[\tau_1], v_2[\tau_2]) \in [\![|[\tau'/\alpha]\tau''|]\!]^{\texttt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

So with the above, the desired result follows by Lemma 8.3 for functions.

Case for $\tau = \exists\beta.\tau''$
In this case we want to show that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{over}_{\exists\beta.\tau''}), \widehat{\delta}_2(\texttt{over}_{[\tau'/\alpha](\exists\beta.\tau'')})) \in [\![\exists\beta.\tau'' \to |[\tau'/\alpha](\exists\beta.\tau'')|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then we assume that $(v_1, v_2) \in [\![\exists\beta.\tau'']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and so by definition we just need to show

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{over}_{\exists\beta.\tau''})\ v_1, \widehat{\delta}_2(\texttt{over}_{\exists\beta.([\tau'/\alpha]\tau'')})\ v_2) \in [\![|\exists\beta.([\tau'/\alpha]\tau'')|]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of over, this is equivalent to

$$(\widehat{\delta}_1([|\tau'|/\alpha](\lambda x : (\exists\beta.\tau'').\,\texttt{unpack}[\beta, y] = x\,\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{\tau''}(y)]\,\texttt{as}\,|\exists\beta.\tau''|))\ v_1),$$
$$\widehat{\delta}_2((\lambda x : (\exists\beta.[\tau'/\alpha]\tau'').\,\texttt{unpack}[\beta, y] = x\,\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{[\tau'/\alpha]\tau''}(y)]\,\texttt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|))\ v_2))$$
$$\in [\![|\exists\beta.([\tau'/\alpha]\tau'')|]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then by Lemma 7.5, this is equivalent to showing

$$(\widehat{\delta}_1(\texttt{unpack}[\beta, y] = v_1\,\texttt{in}(\texttt{pack}[\beta, ([|\tau'|/\alpha]\texttt{over}_{\tau''})(y)]\,\texttt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|)),$$
$$\widehat{\delta}_2(\texttt{unpack}[\beta, y] = v_2\,\texttt{in}(\texttt{pack}[\beta, \texttt{over}_{[\tau'/\alpha]\tau''}(y)]\,\texttt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|)))$$
$$\in [\![|\exists\beta.([\tau'/\alpha]\tau'')|]\!]^{\texttt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Since $(v_1, v_2) \in [\![\exists\beta.\tau'']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$ by assumption, we know by definition that there exist τ_1 type, τ_2 type, e_1 val, e_2 val s.t. $v_1 = \texttt{pack}[\tau_1, e_1]\,\texttt{as}\,\exists\beta.[|\tau'|/\alpha]\tau''$ and $v_2 = \texttt{pack}[\tau_2, e_2]\,\texttt{as}\,\exists\beta.[\tau'/\alpha]\tau''$, as well as a relation $\mathtt{R}'$ such that $(e_1, e_2) \in [\![\tau'']\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$, where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \mathtt{R}'$. Thus we have that equivalently,

$$\begin{aligned}&(\widehat{\delta}_1(\mathtt{unpack}[\beta, y] = \mathtt{pack}[\tau_1, e_1]\,\mathtt{as}\,\exists\beta.[|\tau'|/\alpha]\tau''\,\mathtt{in}(\mathtt{pack}[\beta, ([|\tau'|/\alpha]\mathtt{over}_{\tau''})(y)]\,\mathtt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|)),\\ &\ \widehat{\delta}_2(\mathtt{unpack}[\beta, y] = \mathtt{pack}[\tau_2, e_2]\,\mathtt{as}\,\exists\beta.[\tau'/\alpha]\tau''\,\mathtt{in}(\mathtt{pack}[\beta, \mathtt{over}_{[\tau'/\alpha]\tau''}(y)]\,\mathtt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|)))\\ &\ \in [\![|\exists\beta.([\tau'/\alpha]\tau'')|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1(\mathtt{pack}[\tau_1, ([\tau_1/\beta][|\tau'|/\alpha]\mathtt{over}_{\tau''})(e_1)]\,\mathtt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|),\\ &\ \widehat{\delta}_2(\mathtt{pack}[\tau_2, ([\tau_2/\beta]\mathtt{over}_{[\tau'/\alpha]\tau''})(e_2)]\,\mathtt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|))\\ &\ \in [\![|\exists\beta.([\tau'/\alpha]\tau'')|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

We can then use the types and values from v_1 and v_2 as the ones that exist to make this hold by using Lemma 8.3 for existentials, and use the relation $\mathtt{R}'$ so we just have to show that

$$(\widehat{\delta}_1(([\tau_1/\beta][|\tau'|/\alpha]\mathtt{over}_{\tau''})(e_1)), \widehat{\delta}_2(([\tau_2/\beta]\mathtt{over}_{[\tau'/\alpha]\tau''})(e_2))) \in [\![|[\tau'/\alpha]\tau''|]\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

However by induction we get that

$$(\widehat{\delta}_1([\tau_1/\beta][|\tau'|/\alpha]\mathtt{over}_{\tau''}), \widehat{\delta}_2([\tau_2/\beta]\mathtt{over}_{[\tau'/\alpha]\tau''})) \in [\![\tau'' \to |[\tau'/\alpha]\tau''|]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

And we already know by our assumption as shown earlier that $(e_1, e_2) \in [\![\tau'']\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$, so combining this with what we get by induction, the desired result follows.

Now we want to show the same for `back`, which is that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\exists\beta.\tau''}), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha](\exists\beta.\tau'')})) \in [\![|[\tau'/\alpha](\exists\beta.\tau'')| \to \exists\beta.\tau'']\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Then we assume that $(v_1, v_2) \in [\![|[\tau'/\alpha](\exists\beta.\tau'')|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$, and so by definition we just need to show

$$(\widehat{\delta}_1([|\tau'|/\alpha]\mathtt{back}_{\exists\beta.\tau''})\ v_1, \widehat{\delta}_2(\mathtt{back}_{\exists\beta.([\tau'/\alpha]\tau'')})\ v_2) \in [\![\exists\beta.\tau'']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

By the definition of `back`, this is equivalent to

$$\begin{aligned}&(\widehat{\delta}_1([|\tau'|/\alpha](\lambda x : |\exists\beta.\tau''|.\,\mathtt{unpack}[\beta, y] = x\,\mathtt{in}(\mathtt{pack}[\beta, \mathtt{back}_{\tau''}(y)]\,\mathtt{as}\,\exists\beta.\tau''))\ v_1),\\ &\ \widehat{\delta}_2((\lambda x : |\exists\beta.[\tau'/\alpha]\tau''|.\,\mathtt{unpack}[\beta, y] = x\,\mathtt{in}(\mathtt{pack}[\beta, \mathtt{back}_{[\tau'/\alpha]\tau''}(y)]\,\mathtt{as}\,\exists\beta.[\tau'/\alpha]\tau''))\ v_2))\\ &\ \in [\![\exists\beta.\tau'']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta}_1([|\tau'|/\alpha]\,\mathtt{unpack}[\beta, y] = v_1\,\mathtt{in}(\mathtt{pack}[\beta, \mathtt{back}_{\tau''}(y)]\,\mathtt{as}\,\exists\beta.\tau'')),\\ &\ \widehat{\delta}_2(\mathtt{unpack}[\beta, y] = v_2\,\mathtt{in}(\mathtt{pack}[\beta, \mathtt{back}_{[\tau'/\alpha]\tau''}(y)]\,\mathtt{as}\,\exists\beta.[\tau'/\alpha]\tau'')))\\ &\ \in [\![\exists\beta.\tau'']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Since $(v_1, v_2) \in [\![|\exists\beta.[\tau'/\alpha]\tau''|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$ by assumption, we know by definition that there exist $\tau_1\ \mathtt{type}$, $\tau_2\ \mathtt{type}$, $e_1\ \mathtt{val}$, $e_2\ \mathtt{val}$ s.t. $v_1 = \mathtt{pack}[\tau_1, e_1]\,\mathtt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|$ and

$v_2 = \mathtt{pack}[\tau_2, e_2]\,\mathtt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|$, as well as a relation $\mathtt{R}'$ such that $(e_1, e_2) \in [\![|[\tau'/\alpha]\tau''|]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$, where $\delta_1'' = \delta_1' \otimes \beta \hookrightarrow \tau_1$, $\delta_2'' = \delta_2' \otimes \beta \hookrightarrow \tau_2$, and $\eta'' = \eta' \otimes \beta \hookrightarrow \mathtt{R}'$. Thus we have that equivalently,

$$\begin{aligned}&(\widehat{\delta_1}([|\tau'|/\alpha]\,\mathtt{unpack}[\beta, y] = \mathtt{pack}[\tau_1, e_1]\,\mathtt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|\,\mathtt{in}(\mathtt{pack}[\beta, \mathtt{back}_{\tau''}(y)]\,\mathtt{as}\,\exists\beta.\tau'')),\\ &\widehat{\delta_2}(\mathtt{unpack}[\beta, y] = \mathtt{pack}[\tau_2, e_2]\,\mathtt{as}\,|\exists\beta.[\tau'/\alpha]\tau''|\,\mathtt{in}(\mathtt{pack}[\beta, \mathtt{back}_{[\tau'/\alpha]\tau''}(y)]\,\mathtt{as}\,\exists\beta.[\tau'/\alpha]\tau'')))\\ &\in [\![|\exists\beta.\tau''|]\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

Then by Lemma 7.5, this is equivalent to showing

$$\begin{aligned}&(\widehat{\delta_1}(\mathtt{pack}[\tau_1, ([\tau_1/\beta][|\tau'|/\alpha]\mathtt{back}_{\tau''})(e_1)]\,\mathtt{as}\,\exists\beta.[|\tau'|/\alpha]\tau''),\\ &\widehat{\delta_2}(\mathtt{pack}[\tau_2, ([\tau_2/\beta]\mathtt{back}_{[\tau'/\alpha]\tau''})(e_2)]\,\mathtt{as}\,\exists\beta.[\tau'/\alpha]\tau''))\\ &\in [\![\exists\beta.\tau'']\!]^{\mathtt{E}}_{\eta':\delta_1'\leftrightarrow\delta_2'}\end{aligned}$$

We can then use the types and values from v_1 and v_2 as the ones that exist to make this hold by using Lemma 8.3 for existentials, and use the relation $\mathtt{R}'$ so we just have to show that

$$(\widehat{\delta_1}(([\tau_1/\beta][|\tau'|/\alpha]\mathtt{back}_{\tau''})(e_1)), \widehat{\delta_2}(([\tau_2/\beta]\mathtt{back}_{[\tau'/\alpha]\tau''})(e_2))) \in [\![\tau'']\!]^{\mathtt{E}}_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

However by induction we get that

$$(\widehat{\delta_1}([\tau_1/\beta][|\tau'|/\alpha]\mathtt{back}_{\tau''}), \widehat{\delta_2}([\tau_2/\beta]\mathtt{back}_{[\tau'/\alpha]\tau''})) \in [\![|[\tau'/\alpha]\tau''| \to \tau'']\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$$

And we already know by our assumption as shown earlier that $(e_1, e_2) \in [\![|[\tau'/\alpha]\tau''|]\!]_{\eta'':\delta_1''\leftrightarrow\delta_2''}$, so combining this with what we get by induction, the desired result follows.

□

11.5 Translation Equivalence

Theorem 11.5. If $\Delta; \Gamma \vdash_S e : \tau \rightsquigarrow \bar{e}$, then $\Delta; \Gamma \vdash e \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e}) : \tau$, where $[\texttt{over}_\Gamma/\Gamma] = [\texttt{over}_{\tau_1}(x_1)/x_1]...[\texttt{over}_{\tau_n}(x_n)/x_n]$ for $\Gamma = x_1 : \tau_1, ..., x_n : \tau_n$.

Proof. By induction over the translation rules.

Case for *Runit*
The rule for this case is

$$\frac{}{\Delta; \Gamma \vdash_S \texttt{()} : \texttt{unit} \rightsquigarrow \texttt{()}}\ \textit{Runit}$$

In this case we know that $\Delta; \Gamma \vdash_S \texttt{()} : \texttt{unit} \rightsquigarrow \texttt{()}$. We want to show that $\Delta; \Gamma \vdash \texttt{()} \cong \texttt{back}_{\texttt{unit}}([\texttt{over}_\Gamma/\Gamma]\texttt{()}) : \texttt{unit}$. However, this follows immediately from the fact that $\Delta; \Gamma \vdash \texttt{back}_{\texttt{unit}}([\texttt{over}_\Gamma/\Gamma]\texttt{()}) \cong \texttt{()} : \texttt{unit}$.

Case for *Rvar*
The rule for this case is

$$\frac{\Gamma(x) = \tau}{\Delta; \Gamma \vdash_S x : \tau \rightsquigarrow x}\ \textit{Rvar}$$

In this case we know that $\Delta; \Gamma \vdash_S x : \tau \rightsquigarrow x$. We want to show that $\Delta; \Gamma \vdash x \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]x) : \tau$. However, we know that $\Delta; \Gamma \vdash \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]x) \cong \texttt{back}_\tau(\texttt{over}_\tau(x))$ and by Lemma 11.1 we have that $\Delta; \Gamma \vdash \texttt{back}_\tau(\texttt{over}_\tau(x)) \cong x : \tau$. The desired result follows from transitivity.

Case for *Rint*
The rule for this case is

$$\frac{}{\Delta; \Gamma \vdash_S n : \texttt{int} \rightsquigarrow n}\ \textit{Rint}$$

In this case we know that $\Delta; \Gamma \vdash_S n : \texttt{int} \rightsquigarrow n$. We want to show that

$$\Delta; \Gamma \vdash n \cong \texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]n) : \texttt{int}$$

However, this follows immediately from the fact that $\Delta; \Gamma \vdash \texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]n) \cong n : \texttt{int}$.

Case for *Rintop*
The rule for this case is

$$\frac{\Delta; \Gamma \vdash_S e_1 : \texttt{int} \rightsquigarrow \bar{e_1} \quad \Delta; \Gamma \vdash_S e_2 : \texttt{int} \rightsquigarrow \bar{e_2}}{\Delta; \Gamma \vdash_S e_1\, \texttt{p}\, e_2 : \texttt{int} \rightsquigarrow \bar{e_1}\, \texttt{p}\, \bar{e_2}}\ \textit{Rintop}$$

In this case we know that $\Delta; \Gamma \vdash_S e_1\, \texttt{p}\, e_2 : \texttt{int} \rightsquigarrow e_1'\, \texttt{p}\, e_2'$. We want to show that

$$\Delta; \Gamma \vdash e_1\, \texttt{p}\, e_2 \cong \texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma](e_1'\, \texttt{p}\, e_2')) : \texttt{int}$$

By induction we get that

$$\Delta; \Gamma \vdash e_1 \cong \texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]e_1') : \texttt{int}$$

$$\Delta; \Gamma \vdash e_2 \cong \texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]e_2') : \texttt{int}$$

So by the definition of $\texttt{back}_{\texttt{int}}$, we get the following:

$$\begin{aligned}\Delta;\Gamma\vdash e_1\,\texttt{p}\,e_2 &\cong (\texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]e_1'))\,\texttt{p}(\texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]e_2'))\\ &\cong ((\lambda x:\texttt{int}.x)\ ([\texttt{over}_\Gamma/\Gamma]e_1'))\,\texttt{p}((\lambda x:\texttt{int}.x)\ ([\texttt{over}_\Gamma/\Gamma]e_2'))\\ &\cong ([\texttt{over}_\Gamma/\Gamma]e_1')\,\texttt{p}([\texttt{over}_\Gamma/\Gamma]e_2')\\ &\cong [\texttt{over}_\Gamma/\Gamma](e_1'\,\texttt{p}\,e_2')\\ &\cong (\lambda x:\texttt{int}.x)\ ([\texttt{over}_\Gamma/\Gamma](e_1'\,\texttt{p}\,e_2'))\\ &\cong \texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma](e_1'\,\texttt{p}\,e_2'))\\ &:\texttt{int}\end{aligned}$$

Case for *Rifz*
The rule for this case is

$$\frac{\Delta;\Gamma\vdash_S e_1:\texttt{int}\rightsquigarrow\bar{e_1}\quad \Delta;\Gamma\vdash_S e_2:\tau\rightsquigarrow\bar{e_2}\quad \Delta;\Gamma\vdash_S e_3:\tau\rightsquigarrow\bar{e_3}}{\Delta;\Gamma\vdash_S \texttt{ifz}(e_1,e_2,e_3):\tau\rightsquigarrow\texttt{ifz}(\bar{e_1},\bar{e_2},\bar{e_3})}\ Rifz$$

In this case we know that $\Delta;\Gamma\vdash_S \texttt{ifz}(e_1,e_2,e_3):\tau\rightsquigarrow\texttt{ifz}(e_1',e_2',e_3')$. We want to show that

$$\Delta;\Gamma\vdash\texttt{ifz}(e_1',e_2',e_3')\cong\texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]\texttt{ifz}(e_1',e_2',e_3')):\tau$$

By induction we get that

$$\Delta;\Gamma\vdash e_1\cong\texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]e_1'):\texttt{int}$$

$$\Delta;\Gamma\vdash e_2\cong\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]e_2'):\tau$$

$$\Delta;\Gamma\vdash e_3\cong\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]e_3'):\tau$$

Thus we get the following:

$$\begin{aligned}\Delta;\Gamma\vdash\texttt{ifz}(e_1,e_2,e_3) &\cong \texttt{ifz}(\texttt{back}_{\texttt{int}}([\texttt{over}_\Gamma/\Gamma]e_1'),\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]e_2'),\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]e_3'))\\ &\cong \texttt{ifz}((\lambda x:\texttt{int}.x)\ ([\texttt{over}_\Gamma/\Gamma]e_1'),\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]e_2'),\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]e_3'))\\ &\cong \texttt{ifz}([\texttt{over}_\Gamma/\Gamma]e_1',\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]e_2'),\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]e_3'))\\ &\cong \texttt{back}_\tau(\texttt{ifz}([\texttt{over}_\Gamma/\Gamma]e_1',[\texttt{over}_\Gamma/\Gamma]e_2',[\texttt{over}_\Gamma/\Gamma]e_3'))\\ &\cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\texttt{ifz}(e_1',e_2',e_3'))\\ &:\tau\end{aligned}$$

Case for *Rpair*
The rule for this case is

$$\frac{\Delta;\Gamma\vdash_S e_1:\tau_1\rightsquigarrow\bar{e_1}\quad \Delta;\Gamma\vdash_S e_2:\tau_2\rightsquigarrow\bar{e_2}}{\Delta;\Gamma\vdash_S\langle e_1,e_2\rangle:\tau_1\times\tau_2\rightsquigarrow\langle\bar{e_1},\bar{e_2}\rangle}\ Rpair$$

In this case we know that $\Delta;\Gamma\vdash_S\langle e_1,e_2\rangle:\tau_1\times\tau_2\rightsquigarrow\langle e_1',e_2'\rangle$. We want to show that

$$\Delta;\Gamma\vdash\langle e_1,e_2\rangle\cong\texttt{back}_{\tau_1\times\tau_2}([\texttt{over}_\Gamma/\Gamma]\langle e_1',e_2'\rangle):\tau_1\times\tau_2$$

By induction we get that

$$\Delta;\Gamma\vdash e_1\cong\texttt{back}_{\tau_1}([\texttt{over}_\Gamma/\Gamma]e_1'):\tau_1$$

$$\Delta;\Gamma\vdash e_2\cong\texttt{back}_{\tau_2}([\texttt{over}_\Gamma/\Gamma]e_2'):\tau_2$$

So by the definition of $\mathtt{back}_{\tau_1\times\tau_2}$, we get the following:

$$\begin{aligned}\Delta;\Gamma \vdash \langle e_1, e_2\rangle &\cong \langle \mathtt{back}_{\tau_1}([\mathtt{over}_\Gamma/\Gamma]e_1'), \mathtt{back}_{\tau_2}([\mathtt{over}_\Gamma/\Gamma]e_2')\rangle \\ &\cong \mathtt{back}_{\tau_1\times\tau_2}\langle[\mathtt{over}_\Gamma/\Gamma]e_1', [\mathtt{over}_\Gamma/\Gamma]e_2'\rangle \\ &\cong \mathtt{back}_{\tau_1\times\tau_2}([\mathtt{over}_\Gamma/\Gamma]\langle e_1', e_2'\rangle) \\ &: \tau_1 \times \tau_2\end{aligned}$$

Case for *Rproj*
The rule for this case is

$$\frac{\Delta;\Gamma \vdash_S e : \tau_1 \times \tau_2 \rightsquigarrow \bar{e} \quad i \in \{1,2\}}{\Delta;\Gamma \vdash_S \pi_i e : \tau_i \rightsquigarrow \pi_i \bar{e}}\ Rproj$$

In this case we know that $\Delta;\Gamma \vdash_S \pi_i e : \tau_i \rightsquigarrow \pi_i e'$. We want to show that

$$\Delta;\Gamma \vdash \pi_i e \cong \mathtt{back}_{\tau_i}([\mathtt{over}_\Gamma/\Gamma]\pi_i e') : \tau_i$$

By induction we get that

$$\Delta;\Gamma \vdash e \cong \mathtt{back}_{\tau_1\times\tau_2}([\mathtt{over}_\Gamma/\Gamma]e') : \tau_1 \times \tau_2$$

Thus we get the following:

$$\begin{aligned}\Delta;\Gamma \vdash \pi_i e &\cong \pi_i(\mathtt{back}_{\tau_1\times\tau_2}([\mathtt{over}_\Gamma/\Gamma]e')) \\ &\cong \mathtt{back}_{\tau_i}(\pi_i([\mathtt{over}_\Gamma/\Gamma]e')) \\ &\cong \mathtt{back}_{\tau_i}([\mathtt{over}_\Gamma/\Gamma]\pi_i e') \\ &: \tau_i\end{aligned}$$

Case for *Rtlam*
The rule for this case is

$$\frac{\Delta, \alpha;\Gamma \vdash_S e : \tau \rightsquigarrow \bar{e}}{\Delta;\Gamma \vdash_S \Lambda\alpha.e : \forall\alpha.\tau \rightsquigarrow \Lambda\alpha.\bar{e}}\ Rtlam$$

In this case we know that $\Delta;\Gamma \vdash_S \Lambda\alpha.e : \forall\alpha.\tau \rightsquigarrow \Lambda\alpha.\bar{e}$. We want to show that

$$\Delta;\Gamma \vdash \Lambda\alpha.e \cong \mathtt{back}_{\forall\alpha.\tau}([\mathtt{over}_\Gamma/\Gamma](\Lambda\alpha.\bar{e})) : \forall\alpha.\tau$$

By induction we get that $\Delta, \alpha;\Gamma \vdash e \cong \mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e}) : \tau$, and so by congruence we have that $\Delta;\Gamma \vdash \Lambda\alpha.e \cong \Lambda\alpha.\mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e}) : \forall\alpha.\tau$. We know that

$$\begin{aligned}\Delta;\Gamma \vdash \mathtt{back}_{\forall\alpha.\tau}([\mathtt{over}_\Gamma/\Gamma](\Lambda\alpha.\bar{e})) &\cong \mathtt{back}_{\forall\alpha.\tau}(\Lambda\alpha.([\mathtt{over}_\Gamma/\Gamma]\bar{e})) \\ &\cong (\lambda x : \forall\alpha.\tau.\Lambda\alpha.(\mathtt{back}_\tau(x[\alpha])))\ (\Lambda\alpha.([\mathtt{over}_\Gamma/\Gamma]\bar{e})) \\ &\cong \Lambda\alpha.(\mathtt{back}_\tau((\Lambda\alpha.([\mathtt{over}_\Gamma/\Gamma]\bar{e}))[\alpha])) \\ &\cong \Lambda\alpha.(\mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e})) \\ &\cong \Lambda\alpha.e \\ &: \forall\alpha.\tau\end{aligned}$$

Case for *Rtapp*
The rule for this case is

$$\frac{\Delta;\Gamma \vdash_S e : \forall\alpha.\tau \rightsquigarrow \bar{e} \quad \Delta \vdash_S \tau'\ \mathtt{type}}{\Delta;\Gamma \vdash_S e[\tau'] : [\tau'/\alpha]\tau \rightsquigarrow \bar{e}[|\tau'|]}\ Rtapp$$

In this case we know that $\Delta;\Gamma \vdash_S e[\tau'] : [\tau'/\alpha]\tau \rightsquigarrow \bar{e}[|\tau'|]$. We want to show that

$$\Delta;\Gamma \vdash e[\tau'] \cong \mathtt{back}_{[\tau'/\alpha]\tau}([\mathtt{over}_\Gamma/\Gamma]\bar{e}[|\tau'|]) : [\tau'/\alpha]\tau$$

By induction we get that $\Delta;\Gamma \vdash e \cong \mathtt{back}_{\forall\alpha.\tau}([\mathtt{over}_\Gamma/\Gamma]\bar{e}) : \forall\alpha.\tau$, so by congruence we get that $\Delta;\Gamma \vdash e[\tau'] \cong \mathtt{back}_{\forall\alpha.\tau}([\mathtt{over}_\Gamma/\Gamma]\bar{e})[\tau'] : [\tau'/\alpha]\tau$. We know that

$$\begin{aligned}
\Delta;\Gamma \vdash e[\tau'] &\cong \mathtt{back}_{\forall\alpha.\tau}([\mathtt{over}_\Gamma/\Gamma]\bar{e})[\tau'] \\
&\cong (\lambda x : \forall\alpha.\tau.\Lambda\alpha.(\mathtt{back}_\tau(x[\alpha])) \ [\mathtt{over}_\Gamma/\Gamma]\bar{e})[\tau'] \\
&\cong \Lambda\alpha.(\mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e}[\alpha])) \ [\tau'] \\
&\cong ([\tau'/\alpha]\mathtt{back}_\tau) \ ([\mathtt{over}_\Gamma/\Gamma]\bar{e}[\tau']) \\
&: [\tau'/\alpha]\tau
\end{aligned}$$

By Reflexivity, we know that $\Delta;\Gamma \vdash [\mathtt{over}_\Gamma/\Gamma]\bar{e} \sim [\mathtt{over}_\Gamma/\Gamma]\bar{e} : |\forall\alpha.\tau|$. This means that for $\delta_1 : \Delta$, $\delta_2 : \Delta$, $\eta : \delta_1 \leftrightarrow \delta_2$, and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$, we know that

$$(\widehat{\gamma}_1(\widehat{\delta}_1([\mathtt{over}_\Gamma/\Gamma]\bar{e})), \widehat{\gamma}_2(\widehat{\delta}_2([\mathtt{over}_\Gamma/\Gamma]\bar{e}))) \in [\![|\forall\alpha.\tau|]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

Thus we can use the definition of the logical relation and the extension to terms to get that, using the relation $\mathtt{R}$ as described in Lemma 11.3,

$$(\widehat{\gamma}_1(\widehat{\delta}_1([\mathtt{over}_\Gamma/\Gamma]\bar{e}))[\widehat{\delta}_1(\tau')], \widehat{\gamma}_2(\widehat{\delta}_2([\mathtt{over}_\Gamma/\Gamma]\bar{e}))[\widehat{\delta}_2(|\tau'|)]) \in [\![|\tau|]\!]^{\mathtt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

Where $\delta'_1 = \delta_1 \otimes \alpha \hookrightarrow \widehat{\delta}_1(\tau')$, $\delta'_2 = \delta_2 \otimes \alpha \hookrightarrow \widehat{\delta}_2(|\tau'|)$, and $\eta' = \eta \otimes \alpha \hookrightarrow \mathtt{R}$. By Lemma 11.3 we know that

$$(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_\tau), \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau})) \in [\![|\tau| \to [\tau'/\alpha]\tau]\!]^{\mathtt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}$$

This implies by definition and Lemma 8.3 that

$$\begin{aligned}
&(\widehat{\delta}_1([\tau'/\alpha]\mathtt{back}_\tau) \ (\widehat{\gamma}_1(\widehat{\delta}_1([\mathtt{over}_\Gamma/\Gamma]\bar{e}))[\widehat{\delta}_1(\tau')]), \\
&\ \widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau}) \ (\widehat{\gamma}_2(\widehat{\delta}_2([\mathtt{over}_\Gamma/\Gamma]\bar{e}))[\widehat{\delta}_2(|\tau'|)])) \in [\![[\tau'/\alpha]\tau]\!]^{\mathtt{E}}_{\eta':\delta'_1\leftrightarrow\delta'_2}
\end{aligned}$$

This no longer depends on α, so this is the same as

$$(\widehat{\gamma}_1(\widehat{\delta}_1(([\tau'/\alpha]\mathtt{back}_\tau)([\mathtt{over}_\Gamma/\Gamma]\bar{e}[\tau']))), \widehat{\gamma}_2(\widehat{\delta}_2(\mathtt{back}_{[\tau'/\alpha]\tau}([\mathtt{over}_\Gamma/\Gamma]\bar{e}[|\tau'|])))) \in [\![[\tau'/\alpha]\tau]\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}$$

Then by the definition of logical equivalence, we have that

$$\Delta;\Gamma \vdash ([\tau'/\alpha]\mathtt{back}_\tau)([\mathtt{over}_\Gamma/\Gamma]\bar{e}[\tau']) \sim \mathtt{back}_{[\tau'/\alpha]\tau}([\mathtt{over}_\Gamma/\Gamma]\bar{e}[|\tau'|]) : [\tau'/\alpha]\tau$$

So by the coincidence of logical and contextual equivalence, we get

$$\Delta;\Gamma \vdash ([\tau'/\alpha]\mathtt{back}_\tau)([\mathtt{over}_\Gamma/\Gamma]\bar{e}[\tau']) \cong \mathtt{back}_{[\tau'/\alpha]\tau}([\mathtt{over}_\Gamma/\Gamma]\bar{e}[|\tau'|]) : [\tau'/\alpha]\tau$$

But combining this with our derivation above, this gets us the desired result.

Case for *Rpack*
The rule for this case is

$$\frac{\Delta \vdash_S \tau' \ \mathtt{type} \quad \Delta, \alpha \vdash_S \tau \ \mathtt{type} \quad \Delta;\Gamma \vdash_S e : [\tau'/\alpha]\tau \rightsquigarrow \bar{e}}{\Delta;\Gamma \vdash_S \mathtt{pack}[\tau', e] \ \mathtt{as} \ \exists\alpha.\tau : \exists\alpha.\tau \rightsquigarrow \mathtt{pack}[|\tau'|, \bar{e}] \ \mathtt{as} \ \exists\alpha.|\tau|} \ \textit{Rpack}$$

In this case we know that $\Delta;\Gamma \vdash_S \texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau : \exists\alpha.\tau \rightsquigarrow \texttt{pack}[|\tau'|, \bar{e}]\,\texttt{as}\,\exists\alpha.|\tau|$. We want to show that

$$\Delta;\Gamma \vdash \texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau \cong \texttt{back}_{\exists\alpha.\tau}([\texttt{over}_\Gamma/\Gamma](\texttt{pack}[|\tau'|, \bar{e}]\,\texttt{as}\,\exists\alpha.|\tau|)) : \exists\alpha.\tau$$

By induction we get that $\Delta;\Gamma \vdash e \cong \texttt{back}_{[\tau'/\alpha]\tau}([\texttt{over}_\Gamma/\Gamma]\bar{e}) : [\tau'/\alpha]\tau$, and so by congruence we get that

$$\Delta;\Gamma \vdash \texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau \cong \texttt{pack}[\tau', \texttt{back}_{[\tau'/\alpha]\tau}([\texttt{over}_\Gamma/\Gamma]\bar{e})]\,\texttt{as}\,\exists\alpha.\tau : [\tau'/\alpha]\tau$$

Thus we have that

$$\begin{aligned}
\Delta;\Gamma \vdash\ & \texttt{back}_{\exists\alpha.\tau}([\texttt{over}_\Gamma/\Gamma](\texttt{pack}[|\tau'|, \bar{e}]\,\texttt{as}\,\exists\alpha.|\tau|)) \\
&\cong (\lambda x : |\exists\alpha.\tau|.\,\texttt{unpack}[\alpha, y] = x\,\texttt{in}(\texttt{pack}[\alpha, \texttt{back}_\tau y]\,\texttt{as}\,\exists\alpha.\tau)) \\
&\qquad ([\texttt{over}_\Gamma/\Gamma](\texttt{pack}[|\tau'|, \bar{e}]\,\texttt{as}\,\exists\alpha.|\tau|)) \\
&\cong \texttt{unpack}[\alpha, y] = (\texttt{pack}[|\tau'|, [\texttt{over}_\Gamma/\Gamma]\bar{e}]\,\texttt{as}\,\exists\alpha.|\tau|)\,\texttt{in}(\texttt{pack}[\alpha, \texttt{back}_\tau y]\,\texttt{as}\,\exists\alpha.\tau)) \\
&\cong \texttt{pack}[|\tau'|, ([|\tau'|/\alpha]\texttt{back}_\tau)\ ([\texttt{over}_\Gamma/\Gamma]\bar{e})]\,\texttt{as}\,\exists\alpha.\tau \\
&: \exists\alpha.\tau
\end{aligned}$$

Now we just need to show that

$$\begin{aligned}
\Delta;\Gamma \vdash\ & \texttt{pack}[|\tau'|, ([|\tau'|/\alpha]\texttt{back}_\tau)\ ([\texttt{over}_\Gamma/\Gamma]\bar{e})]\,\texttt{as}\,\exists\alpha.\tau \\
&\cong \texttt{pack}[\tau', \texttt{back}_{[\tau'/\alpha]\tau}([\texttt{over}_\Gamma/\Gamma]\bar{e})]\,\texttt{as}\,\exists\alpha.\tau \\
&: \exists\alpha.\tau
\end{aligned}$$

The desired result will follow from this by our induction result above, as well as transitivity. By the coincidence of contextual and logical equivalence, this is the same as showing that

$$\begin{aligned}
\Delta;\Gamma \vdash\ & \texttt{pack}[|\tau'|, ([|\tau'|/\alpha]\texttt{back}_\tau)\ ([\texttt{over}_\Gamma/\Gamma]\bar{e})]\,\texttt{as}\,\exists\alpha.\tau \\
&\sim \texttt{pack}[\tau', \texttt{back}_{[\tau'/\alpha]\tau}([\texttt{over}_\Gamma/\Gamma]\bar{e})]\,\texttt{as}\,\exists\alpha.\tau \\
&: \exists\alpha.\tau
\end{aligned}$$

So we just have to show for $\delta_1 : \Delta$, $\delta_2 : \Delta$, $\eta : \delta_1 \leftrightarrow \delta_2$, and $\gamma_1 \sim_\Gamma \gamma_2[\eta : \delta_1 \leftrightarrow \delta_2]$ that

$$\begin{aligned}
&(\widehat{\delta}_1(\widehat{\gamma}_1(\texttt{pack}[|\tau'|, ([|\tau'|/\alpha]\texttt{back}_\tau)\ ([\texttt{over}_\Gamma/\Gamma]\bar{e})]\,\texttt{as}\,\exists\alpha.\tau)), \\
&\ \widehat{\delta}_2(\widehat{\gamma}_2(\texttt{pack}[\tau', \texttt{back}_{[\tau'/\alpha]\tau}([\texttt{over}_\Gamma/\Gamma]\bar{e})]\,\texttt{as}\,\exists\alpha.\tau))) \\
&\ \in [\![\exists\alpha.\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}
\end{aligned}$$

By the definition of the type relation, to show this we can just show for some ST-closed relation $\texttt{R} \subseteq \texttt{Val}(\widehat{\delta}_1(|\tau'|)) \times \texttt{Val}(\widehat{\delta}_2(\tau'))$ that

$$(\widehat{\delta}_1(\widehat{\gamma}_1(([|\tau'|/\alpha]\texttt{back}_\tau)\ ([\texttt{over}_\Gamma/\Gamma]\bar{e}))), \widehat{\delta}_2(\widehat{\gamma}_2(\texttt{back}_{[\tau'/\alpha]\tau}([\texttt{over}_\Gamma/\Gamma]\bar{e})))) \in [\![\tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

Where $\delta_1' = \delta_1 \otimes \alpha \hookrightarrow \widehat{\delta}_1(\tau')$, $\delta_2' = \delta_2 \otimes \alpha \hookrightarrow \widehat{\delta}_2(|\tau'|)$, and $\eta' = \eta \otimes \alpha \hookrightarrow \texttt{R}$. Now we pick R to be the relation defined in Lemma 11.4, so by the lemma we know that

$$(\widehat{\delta}_1([|\tau'|/\alpha]\texttt{back}_\tau), \widehat{\delta}_2(\texttt{back}_{[\tau'/\alpha]\tau})) \in [\![|[\tau'/\alpha]\tau| \to \tau]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

We also know by Reflexivity that

$$(\widehat{\delta}_1(\widehat{\gamma}_1([\texttt{over}_\Gamma/\Gamma]\bar{e})), \widehat{\delta}_2(\widehat{\gamma}_2([\texttt{over}_\Gamma/\Gamma]\bar{e}))) \in [\![|[\tau'/\alpha]\tau|]\!]_{\eta':\delta_1'\leftrightarrow\delta_2'}$$

The above two facts combine to get us the desired result, by definition of the type relation.

Case for *Runpack*
The rule for this case is

$$\frac{\Delta;\Gamma \vdash_S e_1 : \exists\alpha.\tau_1 \rightsquigarrow \bar{e_1} \quad \Delta,\alpha;\Gamma,x:\tau_1 \vdash_S e_2 : \tau_2 \rightsquigarrow \bar{e_2} \quad \Delta \vdash_S \tau_2\,\texttt{type}}{\Delta;\Gamma \vdash_S \texttt{unpack}[\alpha,x] = e_1\,\texttt{in}\,e_2 : \tau_2 \rightsquigarrow \texttt{unpack}[\alpha,x] = \bar{e_1}\,\texttt{in}\,\bar{e_2}}\ \textit{Runpack}$$

In this case we know that $\Delta;\Gamma \vdash_S \texttt{unpack}[\alpha,x] = e_1\,\texttt{in}\,e_2 : \tau \rightsquigarrow \texttt{unpack}[\alpha,x] = \bar{e_1}\,\texttt{in}\,\bar{e_2}$. We want to show that

$$\Delta;\Gamma \vdash_S \texttt{unpack}[\alpha,x] = e_1\,\texttt{in}\,e_2 \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma](\texttt{unpack}[\alpha,x] = \bar{e_1}\,\texttt{in}\,\bar{e_2})) : \tau$$

By induction we get that

$$\Delta;\Gamma \vdash e_1 \cong \texttt{back}_{\exists\alpha.\tau'}([\texttt{over}_\Gamma/\Gamma]\bar{e_1}) : \exists\alpha.\tau$$

$$\Delta,\alpha;\Gamma,x:\tau' \vdash e_2 \cong \texttt{back}_\tau([\texttt{over}_{\Gamma,x:\tau'}/\Gamma,x]\bar{e_2}) : \tau$$

Thus we have that

$$\begin{aligned}
\Delta;\Gamma \vdash\ & \texttt{unpack}[\alpha,x] = e_1\,\texttt{in}\,e_2 \\
& \cong \texttt{unpack}[\alpha,x] = \texttt{back}_{\exists\alpha.\tau'}([\texttt{over}_\Gamma/\Gamma]\bar{e_1})\,\texttt{in}\,\texttt{back}_\tau([\texttt{over}_{\Gamma,x:\tau'}/\Gamma,x]\bar{e_2}) \\
& \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma](\texttt{unpack}[\alpha,x] = \texttt{back}_{\exists\alpha.\tau'}(\bar{e_1})\,\texttt{in}[\texttt{over}_{\tau'}(x)/x]\bar{e_2})) \\
& : \tau
\end{aligned}$$

Thus the result will follow by transitivity and congruence if we just show that

$$\Delta;|\Gamma| \vdash \texttt{unpack}[\alpha,x] = \texttt{back}_{\exists\alpha.\tau'}(\bar{e_1})\,\texttt{in}[\texttt{over}_{\tau'}(x)/x]\bar{e_2} \cong \texttt{unpack}[\alpha,x] = \bar{e_1}\,\texttt{in}\,\bar{e_2} : |\tau|$$

If $\bar{e_1}$ does not terminate, then clearly the above holds, as neither side will terminate, so they are contextually equivalent. So suppose $\bar{e_1}\downarrow$, so we have that $\bar{e_1} \mapsto^* \texttt{pack}[\tau_1, v_1]\,\texttt{as}\,\exists\alpha.\tau'$. Then we have that

$$\begin{aligned}
\Delta;|\Gamma| \vdash\ & \texttt{back}_{\exists\alpha.\tau'}(\bar{e_1}) \\
& \cong (\lambda x : |\exists\alpha.\tau'|.\,\texttt{unpack}[\alpha,y] = x\,\texttt{in}(\texttt{pack}[\alpha,\texttt{back}_{\tau'}(y)]\,\texttt{as}\,\exists\alpha.\tau'))\ \bar{e_1} \\
& \cong \texttt{unpack}[\alpha,y] = \bar{e_1}\ \texttt{in}(\texttt{pack}[\alpha,\texttt{back}_{\tau'}(y)]\,\texttt{as}\,\exists\alpha.\tau') \\
& \cong \texttt{unpack}[\alpha,y] = \texttt{pack}[\tau_1,v_1]\,\texttt{as}\,\exists\alpha.\tau'\ \texttt{in}(\texttt{pack}[\alpha,\texttt{back}_{\tau'}(y)]\,\texttt{as}\,\exists\alpha.\tau') \\
& \cong \texttt{pack}[\tau_1,\texttt{back}_{\tau'}(v_1)]\,\texttt{as}\,\exists\alpha.\tau' \\
& : |\tau|
\end{aligned}$$

So then using this and Lemma 11.1 below we have that

$$\begin{aligned}
\Delta;|\Gamma| \vdash\ & \texttt{unpack}[\alpha,x] = \texttt{back}_{\exists\alpha.\tau'}(\bar{e_1})\,\texttt{in}[\texttt{over}_{\tau'}(x)/x]\bar{e_2} \\
& \cong \texttt{unpack}[\alpha,x] = \texttt{pack}[\tau_1,\texttt{back}_{\tau'}(v_1)]\,\texttt{as}\,\exists\alpha.\tau'\ \texttt{in}[\texttt{over}_{\tau'}(x)/x]\bar{e_2} \\
& \cong [\texttt{back}_{\tau'}(v_1)/x][\tau_1/\alpha][\texttt{over}_{\tau'}(x)/x]\bar{e_2} \\
& \cong [\tau_1/\alpha][\texttt{over}_{\tau'}(\texttt{back}_{\tau'}(v_1))/x]\bar{e_2} \\
& \cong [\tau_1/\alpha][v_1/x]\bar{e_2} \\
& \cong \texttt{unpack}[\alpha,x] = \texttt{pack}[\tau_1,v_1]\,\texttt{as}\,\exists\alpha.\tau'\ \texttt{in}\,\bar{e_2} \\
& \cong \texttt{unpack}[\alpha,x] = \bar{e_1}\,\texttt{in}\,\bar{e_2} \\
& : |\tau|
\end{aligned}$$

Thus the desired result follows.

Case for $Rfun$

Given the following abbreviations:

$E = \langle x_1, \langle ... \langle x_{n-1}, x_n\rangle ...\rangle\rangle$

$S = [\pi_1 y/x][\pi_1\pi_2 y/x_1]...[\pi_1\pi_2...\pi_2 y/x_{n-1}][\pi_2...\pi_2 y/x_n]$

$F = \mathtt{pack}[\tau_{env}, \langle f, \pi_2 y\rangle]\ \mathtt{as}\ |\tau \rightarrow \tau'|$

The rule for this case is

$$\frac{\begin{array}{cc}\Gamma = x_1 : \tau_1, ..., x_n : \tau_n & \Delta \vdash_S \tau\ \mathtt{type} \\ \Delta; \Gamma, x : \tau, f : \tau \rightarrow \tau' \vdash_S e : \tau' \rightsquigarrow \bar{e} & \tau_{env} = |\tau_1| \times ... \times |\tau_n|\end{array}}{\begin{array}{c}\Delta; \Gamma \vdash_S \mathtt{fun}\ f(x : \tau).e : \tau \rightarrow \tau' \rightsquigarrow \mathtt{pack}[\tau_{env}, \\ \langle(\widehat{\mathtt{fun}}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})), E\rangle]\ \mathtt{as}\ |\tau \rightarrow \tau'|\end{array}}\ Rfun$$

In this case we have that

$$\Delta; \Gamma \vdash_S \mathtt{fun}\ f(x : \tau).e : \tau \rightarrow \tau' \rightsquigarrow \mathtt{pack}[\tau_{env}, \langle\widehat{\mathtt{fun}}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), E\rangle]\ \mathtt{as}\ |\tau \rightarrow \tau'|$$

We want to show that

$$\begin{aligned}&\Delta; \Gamma \vdash \mathtt{fun}\ f(x : \tau).e : \tau \rightarrow \tau' \\ &\quad \cong \mathtt{back}_{\tau\rightarrow\tau'}([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle\widehat{\mathtt{fun}}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), E\rangle]\ \mathtt{as}\ |\tau \rightarrow \tau'|) \\ &\quad : \tau \rightarrow \tau'\end{aligned}$$

By induction we get that $\Delta; \Gamma, x : \tau, f : \tau \rightarrow \tau' \vdash e \cong \mathtt{back}_{\tau'}([\mathtt{over}_{\Gamma,x:\tau,f:\tau\rightarrow\tau'}/\Gamma, x, f]\bar{e}) : \tau'$. We will prove the above by making use of Admissibility, Theorem 7.8, so we just have to show that for all $i \geq 0$,

$$\begin{aligned}&\Delta; \Gamma \vdash \mathtt{fun}^i\ f(x : \tau).e : \tau \rightarrow \tau' \\ &\quad \cong \mathtt{back}_{\tau\rightarrow\tau'}([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle\widehat{\mathtt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), E\rangle]\ \mathtt{as}\ |\tau \rightarrow \tau'|) \\ &\quad : \tau \rightarrow \tau'\end{aligned}$$

We will do this by induction on i.

Base Case: $i = 0$

In this case we just need to show that

$$\begin{aligned}&\Delta; \Gamma \vdash \mathtt{fun}^0\ f(x : \tau).e \\ &\quad \cong \mathtt{back}_{\tau\rightarrow\tau'}([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle\widehat{\mathtt{fun}}^0\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), E\rangle]\ \mathtt{as}\ |\tau \rightarrow \tau'|) \\ &\quad : \tau \rightarrow \tau'\end{aligned}$$

However we know the following:

$$
\begin{aligned}
\Delta;\Gamma \vdash & \mathtt{back}_{\tau\to\tau'}([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle(\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[F/f]S(\bar{e})), E\rangle]\,\mathtt{as}\,|\tau\to\tau'|)\\
&\cong(\lambda f:|\tau\to\tau'|.\lambda z:\tau.\,\mathtt{unpack}[\alpha,g]=f\,\mathtt{in}\,\mathtt{back}_{\tau'}((\pi_1 g)\hat{}\langle\mathtt{over}_\tau z, \pi_2 g\rangle))\\
&\quad([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle(\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[F/f]S(\bar{e})), E\rangle]\,\mathtt{as}\,|\tau\to\tau'|)\\
&\cong\lambda z:\tau.\,\mathtt{unpack}[\alpha,g]=([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle(\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[F/f]S(\bar{e})),\\
&\qquad E\rangle]\,\mathtt{as}\,|\tau\to\tau'|)\,\mathtt{in}\,\mathtt{back}_{\tau'}((\pi_1 g)\hat{}\langle\mathtt{over}_\tau z, \pi_2 g\rangle)\\
&\cong\lambda z:\tau.\,\mathtt{unpack}[\alpha,g]=(\mathtt{pack}[\tau_{env}, \langle(\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[F/f]S(\bar{e})),\\
&\qquad [\mathtt{over}_\Gamma/\Gamma]E\rangle]\,\mathtt{as}\,|\tau\to\tau'|)\,\mathtt{in}\,\mathtt{back}_{\tau'}((\pi_1 g)\hat{}\langle\mathtt{over}_\tau z, \pi_2 g\rangle)\\
&\cong\lambda z:\tau.\mathtt{back}_{\tau'}((\pi_1\langle(\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[F/f]S(\bar{e})), [\mathtt{over}_\Gamma/\Gamma]E\rangle)\hat{}\\
&\qquad \langle\mathtt{over}_\tau z, \pi_2\langle(\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[F/f]S(\bar{e})), [\mathtt{over}_\Gamma/\Gamma]E\rangle\rangle)\\
&\cong\lambda z:\tau.\mathtt{back}_{\tau'}((\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[F/f]S(\bar{e}))\hat{}\langle\mathtt{over}_\tau z, [\mathtt{over}_\Gamma/\Gamma]E\rangle)\\
&\cong\lambda z:\tau.\mathtt{back}_{\tau'}((\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[\mathtt{pack}[\tau_{env},\langle f, \pi_2 y\rangle]\,\mathtt{as}\,|\tau\to\tau'|/f]S(\bar{e}))\hat{}\\
&\qquad \langle\mathtt{over}_\tau z, [\mathtt{over}_\Gamma/\Gamma]E\rangle)
\end{aligned}
$$

Thus we can just show that

$$
\begin{aligned}
\Delta;\Gamma \vdash\ & \mathtt{fun}^0\ f(x:\tau).e\\
&\sim \lambda z:\tau.\mathtt{back}_{\tau'}((\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[\mathtt{pack}[\tau_{env},\langle f, \pi_2 y\rangle]\,\mathtt{as}\,|\tau\to\tau'|/f]S(\bar{e}))\hat{}\\
&\quad \langle\mathtt{over}_\tau z, [\mathtt{over}_\Gamma/\Gamma]E\rangle)\\
&:\tau\to\tau'
\end{aligned}
$$

So we just have to show for $\delta_1:\Delta$, $\delta_2:\Delta$, $\eta:\delta_1\leftrightarrow\delta_2$, and $\gamma_1 \sim_\Gamma \gamma_2[\eta:\delta_1\leftrightarrow\delta_2]$ that

$$
\begin{aligned}
&(\mathtt{fun}^0\ f(x:\tau).e, \lambda z:\tau.\mathtt{back}_{\tau'}((\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[\mathtt{pack}[\tau_{env},\langle f, \pi_2 y\rangle]\,\mathtt{as}\\
&\quad |\tau\to\tau'|/f]S(\bar{e}))\hat{}\langle\mathtt{over}_\tau z, [\mathtt{over}_\Gamma/\Gamma]E\rangle))\\
&\in [\![\tau\to\tau']\!]_{\eta:\delta_1\leftrightarrow\delta_2}
\end{aligned}
$$

To do this we assume that $(v_1, v_2) \in [\![\tau]\!]_{\eta:\delta_1\leftrightarrow\delta_2}$ and show that

$$
\begin{aligned}
&((\mathtt{fun}^0\ f(x:\tau).e)\ v_1, (\lambda z:\tau.\mathtt{back}_{\tau'}((\widehat{\mathtt{fun}}^0\ f(y:|\tau|\times\tau_{env}).[\mathtt{pack}[\tau_{env},\langle f, \pi_2 y\rangle]\,\mathtt{as}\\
&\quad |\tau\to\tau'|/f]S(\bar{e}))\hat{}\langle\mathtt{over}_\tau z, [\mathtt{over}_\Gamma/\Gamma]E\rangle))\ v_2)\\
&\in [\![\tau']\!]^{\mathtt{E}}_{\eta:\delta_1\leftrightarrow\delta_2}
\end{aligned}
$$

However by the properties of $\mathtt{fun}^0$, we know that neither of these terminate, so the above holds.

Inductive Case: $i > 0$

In this case we just need to show that

$$
\begin{aligned}
\Delta;\Gamma \vdash\ & \mathtt{fun}^i\ f(x:\tau).e\\
&\cong \mathtt{back}_{\tau\to\tau'}([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle\widehat{\mathtt{fun}}^i\ f(y:|\tau|\times\tau_{env}).[F/f]S(\bar{e}), E\rangle]\,\mathtt{as}\,|\tau\to\tau'|)\\
&:\tau\to\tau'
\end{aligned}
$$

Thus we have that

$$
\begin{aligned}
\Delta; \Gamma \vdash & \texttt{back}_{\tau \to \tau'}([\texttt{over}_\Gamma/\Gamma]\,\texttt{pack}[\tau_{env}, \langle(\widehat{\texttt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})), E\rangle]\,\texttt{as}\,|\tau \to \tau'|) \\
& \cong (\lambda f : |\tau \to \tau'|.\lambda z : \tau.\,\texttt{unpack}[\alpha, g] = f\,\texttt{in}\,\texttt{back}_{\tau'}((\pi_1 g)\hat{~}\langle \texttt{over}_\tau z, \pi_2 g\rangle)) \\
& \quad ([\texttt{over}_\Gamma/\Gamma]\,\texttt{pack}[\tau_{env}, \langle(\widehat{\texttt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})), E\rangle]\,\texttt{as}\,|\tau \to \tau'|) \\
& \cong \lambda z : \tau.\,\texttt{unpack}[\alpha, g] = ([\texttt{over}_\Gamma/\Gamma]\,\texttt{pack}[\tau_{env}, \langle(\widehat{\texttt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})), \\
& \qquad E\rangle]\,\texttt{as}\,|\tau \to \tau'|)\,\texttt{in}\,\texttt{back}_{\tau'}((\pi_1 g)\hat{~}\langle \texttt{over}_\tau z, \pi_2 g\rangle) \\
& \cong \lambda z : \tau.\,\texttt{unpack}[\alpha, g] = (\texttt{pack}[\tau_{env}, \langle(\widehat{\texttt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})), \\
& \qquad [\texttt{over}_\Gamma/\Gamma]E\rangle]\,\texttt{as}\,|\tau \to \tau'|)\,\texttt{in}\,\texttt{back}_{\tau'}((\pi_1 g)\hat{~}\langle \texttt{over}_\tau z, \pi_2 g\rangle) \\
& \cong \lambda z : \tau.\texttt{back}_{\tau'}((\pi_1\langle(\widehat{\texttt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})), [\texttt{over}_\Gamma/\Gamma]E\rangle)\hat{~} \\
& \qquad \langle \texttt{over}_\tau z, \pi_2\langle(\widehat{\texttt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})), [\texttt{over}_\Gamma/\Gamma]E\rangle\rangle) \\
& \cong \lambda z : \tau.\texttt{back}_{\tau'}((\widehat{\texttt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}))\hat{~}\langle \texttt{over}_\tau z, [\texttt{over}_\Gamma/\Gamma]E\rangle) \\
& \cong \lambda z : \tau.\texttt{back}_{\tau'}((\widehat{\texttt{fun}}^i\ f(y : |\tau| \times \tau_{env}).[\texttt{pack}[\tau_{env}, \langle f, \pi_2 y\rangle]\,\texttt{as}\,|\tau \to \tau'|/f]S(\bar{e}))\hat{~} \\
& \qquad \langle \texttt{over}_\tau z, [\texttt{over}_\Gamma/\Gamma]E\rangle) \\
& \cong \lambda z : \tau.\texttt{back}_{\tau'}([\widehat{\texttt{fun}}^{i-1}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})/f] \\
& \qquad [\texttt{pack}[\tau_{env}, \langle f, \pi_2\langle \texttt{over}_\tau z, [\texttt{over}_\Gamma/\Gamma]E\rangle\rangle]\,\texttt{as}\,|\tau \to \tau'|/f][\langle \texttt{over}_\tau z, [\texttt{over}_\Gamma/\Gamma]E\rangle/y]S(\bar{e}) \\
& \cong \lambda z : \tau.\texttt{back}_{\tau'}([\widehat{\texttt{fun}}^{i-1}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})/f] \\
& \qquad [\texttt{pack}[\tau_{env}, \langle f, [\texttt{over}_\Gamma/\Gamma]E\rangle]\,\texttt{as}\,|\tau \to \tau'|/f][\langle \texttt{over}_\tau z, [\texttt{over}_\Gamma/\Gamma]E\rangle/y]S(\bar{e}) \\
& \cong \lambda z : \tau.\texttt{back}_{\tau'}([\widehat{\texttt{fun}}^{i-1}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e})/f] \\
& \qquad [\texttt{pack}[\tau_{env}, \langle f, [\texttt{over}_\Gamma/\Gamma]E\rangle]\,\texttt{as}\,|\tau \to \tau'|/f][\texttt{over}_{\Gamma, z:\tau}/\Gamma, z]\bar{e}) \\
& \cong \lambda z : \tau.\texttt{back}_{\tau'}([\texttt{pack}[\tau_{env}, \langle\widehat{\texttt{fun}}^{i-1}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), \\
& \qquad [\texttt{over}_\Gamma/\Gamma]E\rangle]\,\texttt{as}\,|\tau \to \tau'|/f][\texttt{over}_{\Gamma, z:\tau}/\Gamma, z]\bar{e})
\end{aligned}
$$

By our inner induction, we get that

$$
\begin{aligned}
\Delta; \Gamma \vdash\ & \texttt{fun}^{i-1}\ f(x : \tau).e \\
& \cong \texttt{back}_{\tau \to \tau'}([\texttt{over}_\Gamma/\Gamma]\,\texttt{pack}[\tau_{env}, \langle\widehat{\texttt{fun}}^{i-1}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), E\rangle]\,\texttt{as}\,|\tau \to \tau'|) \\
& : \tau \to \tau'
\end{aligned}
$$

However by Lemma 11.1, this is equivalent to

$$
\begin{aligned}
\Delta; \Gamma \vdash\ & \texttt{over}_{\tau \to \tau'}(\texttt{fun}^{i-1}\ f(x : \tau).e) \\
& \cong [\texttt{over}_\Gamma/\Gamma]\,\texttt{pack}[\tau_{env}, \langle\widehat{\texttt{fun}}^{i-1}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), E\rangle]\,\texttt{as}\,|\tau \to \tau'| \\
& \cong \texttt{pack}[\tau_{env}, \langle\widehat{\texttt{fun}}^{i-1}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), [\texttt{over}_\Gamma/\Gamma]E\rangle]\,\texttt{as}\,|\tau \to \tau'| \\
& : \tau \to \tau'
\end{aligned}
$$

We can then apply this to the above to get that

$$\begin{aligned}\Delta;\Gamma \vdash &\lambda z : \tau.\mathtt{back}_{\tau'}([\mathtt{pack}[\tau_{env}, \langle\widehat{\mathtt{fun}}^{i-1}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}),\\ &\quad [\mathtt{over}_\Gamma/\Gamma]E\rangle]\ \mathtt{as}\ |\tau \to \tau'|/f][\mathtt{over}_{\Gamma,z:\tau}/\Gamma, z]\bar{e})\\ &\cong \lambda z : \tau.\mathtt{back}_{\tau'}([\mathtt{over}_{\tau\to\tau'}(\mathtt{fun}^{i-1}\ f(y : \tau).e)/f][\mathtt{over}_{\Gamma,z:\tau}/\Gamma, z]\bar{e})\\ &\cong \lambda z : \tau.\mathtt{back}_{\tau'}([\mathtt{fun}^{i-1}\ f(y : \tau).e/f][\mathtt{over}_{\tau\to\tau'}f/f][\mathtt{over}_{\Gamma,z:\tau}/\Gamma, z]\bar{e})\\ &\cong \lambda z : \tau.[\mathtt{fun}^{i-1}\ f(y : \tau).e/f]\mathtt{back}_{\tau'}([\mathtt{over}_{\Gamma,z:\tau,f:\tau\to\tau'}/\Gamma, z, f]\bar{e})\\ &\cong \lambda z : \tau.[\mathtt{fun}^{i-1}\ f(y : \tau).e/f]e\\ &\cong \mathtt{fun}^{i}\ f(x : \tau).e\\ &\quad : \tau \to \tau'\end{aligned}$$

The second to last equivalence is due to our outer induction hypothesis. Therefore by transitivity this case holds.

Thus we have shown for all i that

$$\begin{aligned}&\Delta;\Gamma \vdash \mathtt{fun}^{i}\ f(x : \tau).e\\ &\quad\cong \mathtt{back}_{\tau\to\tau'}([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle\widehat{\mathtt{fun}}^{i}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), E\rangle]\ \mathtt{as}\ |\tau \to \tau'|)\\ &\quad : \tau \to \tau'\end{aligned}$$

So by Admissibility we get the desired result for this case, that

$$\begin{aligned}&\Delta;\Gamma \vdash \mathtt{fun}\ f(x : \tau).e\\ &\quad\cong \mathtt{back}_{\tau\to\tau'}([\mathtt{over}_\Gamma/\Gamma]\,\mathtt{pack}[\tau_{env}, \langle\widehat{\mathtt{fun}}\ f(y : |\tau| \times \tau_{env}).[F/f]S(\bar{e}), E\rangle]\ \mathtt{as}\ |\tau \to \tau'|)\\ &\quad : \tau \to \tau'\end{aligned}$$

Case for *Rapp*
The rule for this case is

$$\frac{\Delta;\Gamma \vdash_S e_1 : \tau \to \tau' \rightsquigarrow \bar{e_1} \quad \Delta;\Gamma \vdash_S e_2 : \tau \rightsquigarrow \bar{e_2}}{\Delta;\Gamma \vdash_S e_1\ e_2 : \tau' \rightsquigarrow \mathtt{unpack}[\alpha, x] = \bar{e_1}\ \mathtt{in}(\pi_1 x)\hat{\ }\langle\bar{e_2}, \pi_2 x\rangle}\ \textit{Rapp}$$

In this case we have that $\Delta;\Gamma \vdash_S e_1\ e_2 : \tau' \rightsquigarrow \mathtt{unpack}[\alpha, x] = \bar{e_1}\ \mathtt{in}(\pi_1 x)\hat{\ }\langle\bar{e_2}, \pi_2 x\rangle$. We want to show that

$$\Delta;\Gamma \vdash e_1\ e_2 \cong \mathtt{back}_{\tau'}([\mathtt{over}_\Gamma/\Gamma](\mathtt{unpack}[\alpha, x] = \bar{e_1}\ \mathtt{in}(\pi_1 x)\hat{\ }\langle\bar{e_2}, \pi_2 x\rangle)) : \tau'$$

By induction we get that

$$\Delta;\Gamma \vdash e_1 \cong \mathtt{back}_{\tau\to\tau'}([\mathtt{over}_\Gamma/\Gamma]\bar{e_1}) : \tau \to \tau'$$

$$\Delta;\Gamma \vdash e_2 \cong \mathtt{back}_{\tau}([\mathtt{over}_\Gamma/\Gamma]\bar{e_2}) : \tau$$

Thus we have the following:

$$\begin{aligned}
\Delta;\Gamma \vdash e_1\ e_2 \cong & (\texttt{back}_{\tau\to\tau'}([\texttt{over}_\Gamma/\Gamma]\bar{e_1}))\ (\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e_2})) \\
\cong & ((\lambda f : |\tau \to \tau'|. \lambda y : \tau.\, \texttt{unpack}[\alpha, g] = f\ \texttt{in} \\
& \quad \texttt{back}_{\tau'}((\pi_1 g)\hat{}\langle \texttt{over}_\tau y, \pi_2 g\rangle))\ ([\texttt{over}_\Gamma/\Gamma]\bar{e_1})) \\
& (\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e_2})) \\
\cong & (\lambda y : \tau.\, \texttt{unpack}[\alpha, g] = ([\texttt{over}_\Gamma/\Gamma]\bar{e_1})\ \texttt{in} \\
& \quad \texttt{back}_{\tau'}((\pi_1 g)\hat{}\langle \texttt{over}_\tau y, \pi_2 g\rangle))) \\
& (\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e_2})) \\
\cong & \,\texttt{unpack}[\alpha, g] = ([\texttt{over}_\Gamma/\Gamma]\bar{e_1})\ \texttt{in} \\
& \quad \texttt{back}_{\tau'}((\pi_1 g)\hat{}\langle \texttt{over}_\tau(\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e_2})), \pi_2 g\rangle)) \\
\cong & \,\texttt{unpack}[\alpha, g] = ([\texttt{over}_\Gamma/\Gamma]\bar{e_1})\ \texttt{in}\, \texttt{back}_{\tau'}((\pi_1 g)\hat{}\langle [\texttt{over}_\Gamma/\Gamma]\bar{e_2}, \pi_2 g\rangle)) \\
\cong & \texttt{back}_{\tau'}([\texttt{over}_\Gamma/\Gamma](\texttt{unpack}[\alpha, g] = \bar{e_1}\ \texttt{in}(\pi_1 g)\hat{}\langle \bar{e_2}, \pi_2 g\rangle))) \\
& : \tau'
\end{aligned}$$

□

Corollary 11.6. If $\Delta;\Gamma \vdash_S e : \tau \rightsquigarrow \bar{e}$, then $\Delta; |\Gamma| \vdash \bar{e} \cong \texttt{over}_\tau([\texttt{back}_\Gamma/\Gamma]e) : |\tau|$. where $[\texttt{back}_\Gamma/\Gamma] = [\texttt{back}_{\tau_1}(x_1)/x_1]...[\texttt{back}_{\tau_n}(x_n)/x_n]$ for $\Gamma = x_1 : \tau_1, ..., x_n : \tau_n$.

Proof. By Theorem 11.5, we know that

$$\Delta;\Gamma \vdash e \cong \texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e}) : \tau$$

Then by congruence we get that

$$\Delta;\Gamma \vdash \texttt{over}_\tau([\texttt{back}_\Gamma/\Gamma]e) \cong \texttt{over}_\tau([\texttt{back}_\Gamma/\Gamma]\texttt{back}_\tau([\texttt{over}_\Gamma/\Gamma]\bar{e})) : \tau$$

However this is equivalent to

$$\Delta;\Gamma \vdash \texttt{over}_\tau([\texttt{back}_\Gamma/\Gamma]e) \cong \texttt{over}_\tau(\texttt{back}_\tau([\texttt{back}_\Gamma \circ \texttt{over}_\Gamma/\Gamma]\bar{e})) : \tau$$

Then by Lemma 11.1 and Lemma 11.2, we get that

$$\Delta;\Gamma \vdash \texttt{over}_\tau([\texttt{back}_\Gamma/\Gamma]e) \cong \bar{e} : \tau$$

□

12 Erasure

We define an erasure type translation τ° and term translation $\Delta;\Gamma \vdash_C e : \tau \rightsquigarrow^\circ \bar{e}$ that translates terms in the combined language into terms in the source language by simply "erasing" closed functions and converting them to normal functions.

The type translation is defined as follows:

$$\begin{aligned}
\alpha^\circ &= \alpha \\
\texttt{unit}^\circ &= \texttt{unit} \\
\texttt{int}^\circ &= \texttt{int} \\
(\tau_1 \times \tau_2)^\circ &= \tau_1^\circ \times \tau_2^\circ \\
(\tau_1 \to \tau_2)^\circ &= \tau_1^\circ \to \tau_2^\circ \\
(\tau_1 \Rightarrow \tau_2)^\circ &= \tau_1^\circ \to \tau_2^\circ \\
(\forall\alpha.\tau)^\circ &= \forall\alpha.(\tau^\circ) \\
(\exists\alpha.\tau)^\circ &= \exists\alpha.(\tau^\circ)
\end{aligned}$$

We also define Γ° by $\cdot^\circ = \cdot$ and $(\Gamma, x : \tau)^\circ = \Gamma^\circ, x : \tau^\circ$

The term translation is defined as follows:

$$\begin{aligned}
\texttt{()}^\circ &= \texttt{()} \\
x^\circ &= x \\
n^\circ &= n \\
(e_1 \,\texttt{p}\, e_2)^\circ &= e_1^\circ \,\texttt{p}\, e_2^\circ \\
\texttt{ifz}(e_1, e_2, e_3)^\circ &= \texttt{ifz}(e_1^\circ, e_2^\circ, e_3^\circ) \\
\langle e_1, e_2\rangle^\circ &= \langle e_1^\circ, e_2^\circ\rangle \\
(\pi_i e)^\circ &= \pi_i(e^\circ) \\
(\Lambda\alpha.e)^\circ &= \Lambda\alpha.e^\circ \\
(e[\tau])^\circ &= e^\circ[\tau^\circ] \\
(\texttt{pack}[\tau', e] \,\texttt{as}\, \exists\alpha.\tau)^\circ &= \texttt{pack}[\tau'^\circ, e^\circ] \,\texttt{as}\, \exists\alpha.\tau^\circ \\
(\texttt{unpack}[\alpha, x] = e_1 \,\texttt{in}\, e_2)^\circ &= \texttt{unpack}[\alpha, x] = e_1^\circ \,\texttt{in}\, e_2^\circ \\
(\texttt{fun}\ f(x : \tau).e)^\circ &= \texttt{fun}\ f(x : \tau^\circ).e^\circ \\
(e_1\ e_2)^\circ &= e_1^\circ\ e_2^\circ \\
(\widehat{\texttt{fun}}\ f(x : \tau).e)^\circ &= \texttt{fun}\ f(x : \tau^\circ).e^\circ \\
(e_1 \,\widehat{}\, e_2)^\circ &= e_1^\circ\ e_2^\circ
\end{aligned}$$

12.1 Static Erasure

Lemma 12.1. If $\Delta;\Gamma \vdash_C e : \tau$, then $\Delta;\Gamma^\circ \vdash_S e^\circ : \tau^\circ$.

Proof. By induction on the structure of e.

Case for $Tunit$
If $\Delta;\Gamma \vdash_C \texttt{()} : \texttt{unit}$, then we know that $\texttt{()}^\circ = \texttt{()}$. But we also know that $\Delta;\Gamma^\circ \vdash_S \texttt{()} : \texttt{unit}^\circ = \texttt{unit}$ by rule $Tunit$, as desired.

Case for $Tvar$
If $\Delta;\Gamma \vdash_C x : \tau$, then we know that $x^\circ = x$. But we also know that $\Delta;\Gamma^\circ \vdash_S x : \tau^\circ$ by rule $Tvar$, since by definition, Γ° will contain $x : \tau^\circ$, since Γ contains $x : \tau$.

Case for $Tint$
If $\Delta;\Gamma \vdash_C n : \texttt{int}$, then we know that $n^\circ = n$. But we also know that $\Delta;\Gamma^\circ \vdash_S n : \texttt{int}^\circ = \texttt{int}$ by rule $Tint$, as desired.

Case for $Tintop$
If $\Delta;\Gamma \vdash_C e_1 \, \mathbf{p} \, e_2 : \texttt{int}$, then we know by assumption that $\Delta;\Gamma \vdash_C e_1 : \texttt{int}$ and $\Delta;\Gamma \vdash_C e_2 : \texttt{int}$. Then by induction we have that $\Delta;\Gamma^\circ \vdash_S e_1^\circ : \texttt{int}$ and $\Delta;\Gamma^\circ \vdash_S e_2^\circ : \texttt{int}$. Thus by rule $Tintop$ and the fact that $(e_1 \, \mathbf{p} \, e_2)^\circ = (e_1^\circ \, \mathbf{p} \, e_2^\circ)$, we get that $\Delta;\Gamma \vdash_S (e_1 \, \mathbf{p} \, e_2)^\circ : \texttt{int}$, as desired.

Case for $Tifz$
If $\Delta;\Gamma \vdash_C \texttt{ifz}(e_1, e_2, e_3) : \tau$, then we know by assumption that $\Delta;\Gamma \vdash_C e_1 : \texttt{int}$, $\Delta;\Gamma \vdash_C e_2 : \tau$, and $\Delta;\Gamma \vdash_C e_3 : \tau$. Then by induction we have that $\Delta;\Gamma^\circ \vdash_S e_1^\circ : \texttt{int}$, $\Delta;\Gamma^\circ \vdash_S e_2^\circ : \tau^\circ$, and $\Delta;\Gamma^\circ \vdash_S e_3^\circ : \tau^\circ$. Thus by rule $Tifz$ and the fact that $\texttt{ifz}(e_1, e_2, e_3)^\circ = \texttt{ifz}(e_1^\circ, e_2^\circ, e_3^\circ)$, we get that $\Delta;\Gamma \vdash_S \texttt{ifz}(e_1, e_2, e_3)^\circ : \tau^\circ$, as desired.

Case for $Tpair$
If $\Delta;\Gamma \vdash_C \langle e_1, e_2 \rangle : \tau_1 \times \tau_2$, then we know by assumption that $\Delta;\Gamma \vdash_C e_1 : \tau_1$ and $\Delta;\Gamma \vdash_C e_2 : \tau_2$. Then by induction we have that $\Delta;\Gamma^\circ \vdash_S e_1^\circ : \tau_1^\circ$ and $\Delta;\Gamma^\circ \vdash_S e_2^\circ : \tau_2^\circ$. Thus by rule $Tpair$ and the fact that $\langle e_1, e_2 \rangle^\circ = \langle e_1^\circ, e_2^\circ \rangle$, we get that $\Delta;\Gamma \vdash_S \langle e_1, e_2 \rangle^\circ : (\tau_1 \times \tau_2)^\circ$, as desired.

Case for $Tproj$
If $\Delta;\Gamma \vdash_C \pi_i e : \tau_i$, then we know by assumption that $\Delta;\Gamma \vdash_C e : \tau_1 \times \tau_2$. Then by induction we have that $\Delta;\Gamma^\circ \vdash_S e^\circ : (\tau_1 \times \tau_2)^\circ$. Thus by rule $Tproj$ and the fact that $(\pi_i e)^\circ = \pi_i(e^\circ)$, we get that $\Delta;\Gamma \vdash_S (\pi_i e)^\circ : \tau_i^\circ$, as desired.

Case for $Ttlam$
If $\Delta;\Gamma \vdash_C \Lambda\alpha.e : \forall\alpha.\tau$, then we know by assumption that $\Delta, \alpha;\Gamma \vdash_C e : \tau$. Then by induction we have that $\Delta, \alpha;\Gamma^\circ \vdash_S e^\circ : \tau^\circ$. Thus by rule $Ttlam$ and the fact that $(\Lambda\alpha.e)^\circ = \Lambda\alpha.(e^\circ)$, we get that $\Delta;\Gamma \vdash_S (\Lambda\alpha.e)^\circ : (\forall\alpha.\tau)^\circ$, as desired.

Case for $Ttapp$
If $\Delta;\Gamma \vdash_C e[\tau'] : [\tau'/\alpha]\tau$, then we know by assumption that $\Delta;\Gamma \vdash_C e : \forall\alpha.\tau$. Then by induction we have that $\Delta;\Gamma^\circ \vdash_S e^\circ : (\forall\alpha.\tau)^\circ$. Thus by rule $Ttapp$ and the fact that $(e[\tau'])^\circ = (e^\circ)[\tau'^\circ]$, we get that $\Delta;\Gamma \vdash_S (e[\tau'])^\circ : ([\tau'/\alpha]\tau)^\circ$, as desired.

Case for $Tpack$
If $\Delta;\Gamma \vdash_C \texttt{pack}[\tau', e] \, \texttt{as} \, \exists\alpha.\tau : \exists\alpha.\tau$, then we know by assumption that $\Delta;\Gamma \vdash_C e : [\tau'/\alpha]\tau$. Then by induction we have that $\Delta;\Gamma^\circ \vdash_S e^\circ : ([\tau'/\alpha]\tau)^\circ$. Thus by rule $Tpack$ and the fact that $(\texttt{pack}[\tau', e] \, \texttt{as} \, \exists\alpha.\tau)^\circ = \texttt{pack}[\tau'^\circ, e^\circ] \, \texttt{as} \, \exists\alpha.\tau^\circ$, we get that $\Delta;\Gamma \vdash_S (\texttt{pack}[\tau', e] \, \texttt{as} \, \exists\alpha.\tau)^\circ : (\exists\alpha.\tau)^\circ$, as desired.

Case for *Tunpack*
If $\Delta; \Gamma \vdash_C \texttt{unpack}[\alpha, x] = e_1 \,\texttt{in}\, e_2 : \tau$, then we know by assumption that $\Delta; \Gamma \vdash_C e_1 : \exists \alpha.\tau'$ and $\Delta, \alpha; \Gamma, x : \tau' \vdash_C e_2 : \tau$. Then by induction we have that $\Delta; \Gamma^\circ \vdash_C e_1^\circ : \exists \alpha.\tau'^\circ$ and $\Delta, \alpha; \Gamma^\circ, x : \tau'^\circ \vdash_C e_2^\circ : \tau^\circ$. Thus by rule *Tunpack* and and the fact that $(\texttt{unpack}[\alpha, x] = e_1 \,\texttt{in}\, e_2)^\circ = \texttt{unpack}[\alpha, x] = e_1^\circ \,\texttt{in}\, e_2^\circ$, we get that $\Delta; \Gamma \vdash_S (\texttt{unpack}[\alpha, x] = e_1 \,\texttt{in}\, e_2)^\circ : \tau^\circ$, as desired.

Case for $Tfun$
If $\Delta; \Gamma \vdash_C \texttt{fun}\ f(x : \tau).e : \tau \to \tau'$, then we know by assumption that $\Delta; \Gamma, x : \tau \vdash_C e : \tau'$. Then by induction we have that $\Delta; \Gamma^\circ, x : \tau^\circ \vdash_S e^\circ : \tau'^\circ$. Thus by rule $Tfun$ and the fact that $(\texttt{fun}\ f(x : \tau).e)^\circ = \texttt{fun}\ f(x : \tau^\circ).e^\circ$, we get that $\Delta; \Gamma \vdash_S (\texttt{fun}\ f(x : \tau).e)^\circ : (\tau \to \tau')^\circ$, as desired.

Case for $Tapp$
If $\Delta; \Gamma \vdash_C e_1\ e_2 : \tau'$, then we know by assumption that $\Delta; \Gamma \vdash_C e_1 : \tau \to \tau'$ and $\Delta; \Gamma \vdash_C e_2 : \tau$. Then by induction we have that $\Delta; \Gamma^\circ \vdash_S e_1^\circ : (\tau \to \tau')^\circ$ and $\Delta; \Gamma^\circ \vdash_S e_2^\circ : \tau^\circ$. Thus by rule $Tapp$ and the fact that $(e_1\ e_2)^\circ = e_1^\circ\ e_2^\circ$, we get that $\Delta; \Gamma \vdash_S (e_1\ e_2)^\circ : \tau'^\circ$, as desired.

Case for $Tcc\widehat{fun}$
If $\Delta; \Gamma \vdash_C \widehat{\texttt{fun}}\ f(x : \tau).e : \tau \Rightarrow \tau'$, then we know by assumption that $\Delta; \Gamma, x : \tau \vdash_C e : \tau'$. Then by induction we have that $\Delta; \Gamma^\circ, x : \tau^\circ \vdash_S e^\circ : \tau'^\circ$. Thus by rule $Tfun$ and the fact that $(\widehat{\texttt{fun}}\ f(x : \tau).e)^\circ = \texttt{fun}\ f(x : \tau^\circ).e^\circ$, we get that $\Delta; \Gamma \vdash_S (\widehat{\texttt{fun}}\ f(x : \tau).e)^\circ : (\tau \to \tau')^\circ$, as desired.

Case for $Tccapp$
If $\Delta; \Gamma \vdash_C e_1 \hat{\ } e_2 : \tau'$, then we know by assumption that $\Delta; \Gamma \vdash_C e_1 : \tau \Rightarrow \tau'$ and $\Delta; \Gamma \vdash_C e_2 : \tau$. Then by induction we have that $\Delta; \Gamma^\circ \vdash_S e_1^\circ : (\tau \Rightarrow \tau')^\circ$ and $\Delta; \Gamma^\circ \vdash_S e_2^\circ : \tau^\circ$. Thus by rule $Tapp$ and the fact that $(e_1 \hat{\ } e_2)^\circ = e_1^\circ\ e_2^\circ$, we get that $\Delta; \Gamma \vdash_S (e_1 \hat{\ } e_2)^\circ : \tau'^\circ$, as desired.

□

Lemma 12.2. If $e\ \texttt{val}$, then $e^\circ\ \texttt{val}$.

Proof. By induction on the structure of the step $e \mapsto e'$.

Case for Vunit
Trivial, since $\texttt{()}^\circ = \texttt{()}$.

Case for Vint
Trivial, since $n^\circ = n$.

Case for Vpair
Suppose $\langle e_1, e_2 \rangle\ \texttt{val}$, we want to show that $\langle e_1, e_2 \rangle^\circ\ \texttt{val}$. By induction we get that $e_1^\circ\ \texttt{val}$ and $e_2^\circ\ \texttt{val}$, which implies by rule *Vpair* that $\langle e_1^\circ, e_2^\circ \rangle\ \texttt{val}$, or equivalently that $\langle e_1, e_2 \rangle^\circ\ \texttt{val}$.

Case for Vfun
Suppose $\texttt{fun}\ f(x : \tau).e\ \texttt{val}$, we want to show that $(\texttt{fun}\ f(x : \tau).e)^\circ\ \texttt{val}$. Since $(\texttt{fun}\ f(x : \tau).e)^\circ = \texttt{fun}\ f(x : \tau^\circ).e^\circ$ and $\texttt{fun}\ f(x : \tau^\circ).e^\circ\ \texttt{val}$ by rule $Vfun$, we have that $(\texttt{fun}\ f(x : \tau).e)^\circ\ \texttt{val}$.

Case for Vccfun
Suppose $\widehat{\texttt{fun}}\ f(x : \tau).e\ \texttt{val}$, we want to show that $(\widehat{\texttt{fun}}\ f(x : \tau).e)^\circ\ \texttt{val}$. Since $(\widehat{\texttt{fun}}\ f(x : \tau).e)^\circ = \texttt{fun}\ f(x : \tau^\circ).e^\circ$ and $\texttt{fun}\ f(x : \tau^\circ).e^\circ\ \texttt{val}$ by rule $Vfun$, we have that $(\widehat{\texttt{fun}}\ f(x : \tau).e)^\circ\ \texttt{val}$.

Case for Vtlam
Suppose $\Lambda\alpha.e\,\texttt{val}$, we want to show that $(\Lambda\alpha.e)^\circ\,\texttt{val}$. Since $(\Lambda\alpha.e)^\circ = \Lambda\alpha.e^\circ$, by rule *Vtlam* we know that $\Lambda\alpha.e^\circ\,\texttt{val}$, or equivalently that $(\Lambda\alpha.e)^\circ\,\texttt{val}$.

Case for Vpack
Suppose $\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau\,\texttt{val}$, we want to show that $(\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau)^\circ\,\texttt{val}$. By induction we get that $e^\circ\,\texttt{val}$. Then by rule *Vpack* we get that $\texttt{pack}[\tau'^\circ, e^\circ]\,\texttt{as}\,\exists\alpha.\tau^\circ\,\texttt{val}$, or equivalently that $(\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau)^\circ\,\texttt{val}$.

□

12.2 Dynamic Erasure

Lemma 12.3. If $e \mapsto e'$, then $e^\circ \mapsto e'^\circ$.

Proof. By induction on the structure of the step $e \mapsto e'$.

Case for *Eintop*$_1$
Suppose that

$$\frac{e_1 \mapsto e_1'}{e_1\,\texttt{p}\,e_2 \mapsto e_1'\,\texttt{p}\,e_2}\;Eintop_1$$

Then by induction we get that $e_1^\circ \mapsto e_1'^\circ$, so by rule *Eintop*$_1$ we know that $e_1^\circ\,\texttt{p}\,e_2^\circ \mapsto e_1'^\circ\,\texttt{p}\,e_2^\circ$, which by definition is equivalent to $(e_1\,\texttt{p}\,e_2)^\circ \mapsto (e_1'\,\texttt{p}\,e_2)^\circ$.

Case for *Eintop*$_2$
Suppose that

$$\frac{e_2 \mapsto e_2'}{e_1\,\texttt{p}\,e_2 \mapsto e_1\,\texttt{p}\,e_2'}\;Eintop_2$$

Then by induction we get that $e_2^\circ \mapsto e_2'^\circ$, so by rule *Eintop*$_2$ we know that $e_1^\circ\,\texttt{p}\,e_2^\circ \mapsto e_1^\circ\,\texttt{p}\,e_2'^\circ$, which by definition is equivalent to $(e_1\,\texttt{p}\,e_2)^\circ \mapsto (e_1\,\texttt{p}\,e_2')^\circ$.

Case for *Eintop*$_3$
Suppose that

$$\frac{n_1\,\texttt{p}\,n_2 = n}{n_1\,\texttt{p}\,n_2 \mapsto n}\;Eintop_3$$

Since $n_1^\circ = n_1$ and $n_2^\circ = n_2$ by definition, it follows that $(n_1\,\texttt{p}\,n_2)^\circ = n = n^\circ$, so by rule *Eintop*$_3$ we have that $(n_1\,\texttt{p}\,n_2)^\circ \mapsto n^\circ$.

Case for *Eifz*$_1$
Suppose that

$$\frac{e_1 \mapsto e_1'}{\texttt{ifz}(e_1, e_2, e_3) \mapsto \texttt{ifz}(e_1', e_2, e_3)}\;Eifz_1$$

Then by induction we get that $e_1^\circ \mapsto e_1'^\circ$, so by rule *Eifz*$_1$ we know that $\texttt{ifz}(e_1^\circ, e_2^\circ, e_3^\circ) \mapsto \texttt{ifz}(e_1'^\circ, e_2^\circ, e_3^\circ)$, which by definition is equivalent to $\texttt{ifz}(e_1, e_2, e_3)^\circ \mapsto \texttt{ifz}(e_1', e_2, e_3)^\circ$,

Case for $Eifz_2$
Suppose that

$$\frac{n = 0}{\texttt{ifz}(n, e_2, e_3) \mapsto e_2}\ Eifz_2$$

Since $n^\circ = n = 0$, we know by rule $Eifz_2$ that $\texttt{ifz}(n^\circ, e_2^\circ, e_3^\circ) \mapsto e_2^\circ$ or equivalently $\texttt{ifz}(n, e_2, e_3)^\circ \mapsto e_2^\circ$.

Case for $Eifz_3$
Suppose that

$$\frac{n \neq 0}{\texttt{ifz}(n, e_2, e_3) \mapsto e_3}\ Eifz_3$$

Since $n^\circ = n \neq 0$, we know by rule $Eifz_3$ that $\texttt{ifz}(n^\circ, e_2^\circ, e_3^\circ) \mapsto e_3^\circ$ or equivalently $\texttt{ifz}(n, e_2, e_3)^\circ \mapsto e_3^\circ$.

Case for $Eapp_1$
Suppose that

$$\frac{e_1 \mapsto e_1'}{e_1\ e_2 \mapsto e_1'\ e_2}\ Eapp_1$$

Then by induction we get that $e_1^\circ \mapsto e_1'^\circ$, so by rule $Eapp_1$ we know that $e_1^\circ\ e_2^\circ \mapsto e_1'^\circ\ e_2^\circ$, or equivalently that $(e_1\ e_2)^\circ \mapsto (e_1'\ e_2)^\circ$.

Case for $Eapp_2$
Suppose that

$$\frac{e_2 \mapsto e_2'}{(\texttt{fun}\ f(x : \tau).e)\ e_2 \mapsto (\texttt{fun}\ f(x : \tau).e)\ e_2'}\ Eapp_2$$

Then by induction we get that $e_2^\circ \mapsto e_2'^\circ$, so by rule $Eapp_2$ we know that $(\texttt{fun}\ f(x : \tau).e)^\circ\ e_2^\circ \mapsto (\texttt{fun}\ f(x : \tau).e)^\circ\ e_2'^\circ$, or equivalently that $((\texttt{fun}\ f(x : \tau).e)\ e_2)^\circ \mapsto ((\texttt{fun}\ f(x : \tau).e)\ e_2')^\circ$.

Case for $Eapp_3$
Suppose that

$$\frac{e_2\ \texttt{val}}{(\texttt{fun}\ f(x : \tau).e)\ e_2 \mapsto [\texttt{fun}\ f(x : \tau).e/f][e_2/x]e}\ Eapp_3$$

We know that $(\texttt{fun}\ f(x : \tau).e)^\circ = (\texttt{fun}\ f(x : \tau^\circ).e^\circ)$, as well as that $e_2^\circ\ \texttt{val}$ by Lemma 12.2. Then by rule $Eapp_3$ we get that $(\texttt{fun}\ f(x : \tau^\circ).e^\circ)\ e_2^\circ \mapsto [\texttt{fun}\ f(x : \tau^\circ).e^\circ/f][e_2^\circ/x]e^\circ$ which is equivalent to $((\texttt{fun}\ f(x : \tau).e)\ e_2)^\circ \mapsto ([\texttt{fun}\ f(x : \tau).e/f][e_2/x]e)^\circ$.

Case for $Eccapp_1$
Suppose that

$$\frac{e_1 \mapsto e_1'}{e_1 \widehat{\ } e_2 \mapsto e_1' \widehat{\ } e_2}\ Eccapp_1$$

Then by induction we get that $e_1^\circ \mapsto e_1'^\circ$, so by rule $Eapp_1$ we know that $e_1^\circ\ e_2^\circ \mapsto e_1'^\circ\ e_2^\circ$. Since $(e_1 \widehat{\ } e_2)^\circ = e_1^\circ\ e_2^\circ$ and $(e_1' \widehat{\ } e_2)^\circ = e_1'^\circ\ e_2^\circ$, this is equivalent to $(e_1 \widehat{\ } e_2)^\circ \mapsto (e_1' \widehat{\ } e_2)^\circ$.

Case for $Eccapp_2$
Suppose that

$$\frac{e_2 \mapsto e_2'}{(\widehat{\texttt{fun}}\ f(x : \tau).e) \widehat{\ } e_2 \mapsto (\widehat{\texttt{fun}}\ f(x : \tau).e) \widehat{\ } e_2'}\ Eccapp_2$$

Then by induction we get that $e_2^\circ \mapsto e_2'^\circ$, so by rule *Eapp*$_2$ we know that $(\widehat{\texttt{fun}}\ f(x:\tau).e)^\circ\ e_2^\circ \mapsto (\widehat{\texttt{fun}}\ f(x:\tau).e)^\circ\ e_2'^\circ$. Since $((\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{\ }e_2)^\circ = (\widehat{\texttt{fun}}\ f(x:\tau).e)^\circ\ e_2^\circ$ and $((\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{\ }e_2')^\circ = (\widehat{\texttt{fun}}\ f(x:\tau).e)^\circ\ e_2'^\circ$, this is equivalent to $((\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{\ }e_2)^\circ \mapsto ((\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{\ }e_2')^\circ$.

Case for *Eccapp*$_3$
Suppose that

$$\frac{e_2\ \texttt{val}}{(\widehat{\texttt{fun}}\ f(x:\tau).e)\hat{\ }e_2 \mapsto [\widehat{\texttt{fun}}\ f(x:\tau).e/f][e_2/x]e}\ \textit{Eccapp}_3$$

We know that $(\widehat{\texttt{fun}}\ f(x:\tau).e)^\circ = (\texttt{fun}\ f(x:\tau^\circ).e^\circ)$, as well as that $e_2^\circ\ \texttt{val}$ by Lemma 12.2. Then by rule *Eapp*$_3$ we get that $(\texttt{fun}\ f(x:\tau^\circ).e^\circ)\ e_2^\circ \mapsto [\texttt{fun}\ f(x:\tau^\circ).e^\circ/f][e_2^\circ/x]e^\circ$ which is equivalent to $((\widehat{\texttt{fun}}\ f(x:\tau).e)\ e_2)^\circ \mapsto ([\widehat{\texttt{fun}}\ f(x:\tau).e/f][e_2/x]e)^\circ$.

Case for *Epair*$_1$
Suppose that

$$\frac{e_1 \mapsto e_1'}{\langle e_1, e_2\rangle \mapsto \langle e_1', e_2\rangle}\ \textit{Epair}_1$$

Then by induction we get that $e_1^\circ \mapsto e_1'^\circ$, so by rule *Epair*$_1$ we know that $\langle e_1^\circ, e_2^\circ\rangle \mapsto \langle e_1'^\circ, e_2^\circ\rangle$, or equivalently that $\langle e_1, e_2\rangle^\circ \mapsto \langle e_1', e_2\rangle^\circ$

Case for *Epair*$_2$
Suppose that

$$\frac{e_1\ \texttt{val} \quad e_2 \mapsto e_2'}{\langle e_1, e_2\rangle \mapsto \langle e_1, e_2'\rangle}\ \textit{Epair}_2$$

Then by induction we get that $e_2^\circ \mapsto e_2'^\circ$, so by rule *Epair*$_1$ we know that $\langle e_1^\circ, e_2^\circ\rangle \mapsto \langle e_1^\circ, e_2'^\circ\rangle$, or equivalently that $\langle e_1, e_2\rangle^\circ \mapsto \langle e_1, e_2'\rangle^\circ$

Case for *Eproj*$_1$
Suppose that

$$\frac{e \mapsto e'}{\pi_i e \mapsto \pi_i e'}\ \textit{Eproj}_1$$

Then by induction we get that $e^\circ \mapsto e'^\circ$, so by rule *Eproj*$_1$ we know that $\pi_i(e^\circ) \mapsto \pi_i(e'^\circ)$, or equivalently that $(\pi_i e)^\circ \mapsto (\pi_i e')^\circ$.

Case for *Eproj*$_2$
Suppose that

$$\frac{i \in \{1,2\} \quad e_1\ \texttt{val} \quad e_2\ \texttt{val}}{\pi_i\langle e_1, e_2\rangle \mapsto e_i}\ \textit{Eproj}_2$$

By Lemma 12.2 we have that $e_1^\circ\ \texttt{val}$ and $e_2^\circ\ \texttt{val}$. Thus by rule *Eproj*$_2$ we have that $\pi_i\langle e_1^\circ, e_2^\circ\rangle \mapsto e_i^\circ$, or equivalently $\pi_i\langle e_1, e_2\rangle^\circ \mapsto e_i^\circ$.

Case for *Etapp*$_1$
Suppose that

$$\frac{e \mapsto e'}{e[\tau] \mapsto e'[\tau]}\ \textit{Etapp}_1$$

Then by induction we get that $e^\circ \mapsto e'^\circ$, so by rule *Etapp*$_1$ we know that $e^\circ[\tau^\circ] \mapsto e'^\circ[\tau^\circ]$, or equivalently that $(e[\tau])^\circ \mapsto (e'[\tau])^\circ$.

Case for *Etapp*$_2$
Suppose that

$$\frac{}{(\Lambda\alpha.e)[\tau] \mapsto [\tau/\alpha]e}\ Etapp_2$$

We know by definition that $(\Lambda\alpha.e)^\circ = \Lambda\alpha.e^\circ$, and we have by rule *Etapp*$_2$ that $(\Lambda\alpha.e^\circ)[\tau^\circ] \mapsto [\tau^\circ/\alpha]e^\circ$, so equivalently $((\Lambda\alpha.e)[\tau])^\circ \mapsto ([\tau/\alpha]e)^\circ$.

Case for *Epack*
Suppose that

$$\frac{e \mapsto e'}{\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau \mapsto \texttt{pack}[\tau', e']\,\texttt{as}\,\exists\alpha.\tau}\ Epack$$

Then by induction we get that $e^\circ \mapsto e'^\circ$, so by rule *Epack* we know that $\texttt{pack}[\tau'^\circ, e^\circ]\,\texttt{as}\,\exists\alpha.\tau^\circ \mapsto \texttt{pack}[\tau'^\circ, e'^\circ]\,\texttt{as}\,\exists\alpha.\tau^\circ$, which is equivalent to $(\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau)^\circ \mapsto (\texttt{pack}[\tau', e']\,\texttt{as}\,\exists\alpha.\tau)^\circ$.

Case for *Eunpack*$_1$
Suppose that

$$\frac{e_1 \mapsto e_1'}{\texttt{unpack}[\alpha, x] = e_1\,\texttt{in}\,e_2 \mapsto \texttt{unpack}[\alpha, x] = e_1'\,\texttt{in}\,e_2}\ Eunpack_1$$

Then by induction we get that $e_1^\circ \mapsto e_1'^\circ$, so by rule *Eunpack*$_1$ we know $\texttt{unpack}[\alpha, x] = e_1^\circ\,\texttt{in}\,e_2^\circ \mapsto \texttt{unpack}[\alpha, x] = e_1'^\circ\,\texttt{in}\,e_2^\circ$, or equivalently that

$$(\texttt{unpack}[\alpha, x] = e_1\,\texttt{in}\,e_2)^\circ \mapsto (\texttt{unpack}[\alpha, x] = e_1'\,\texttt{in}\,e_2)^\circ$$

Case for *Eunpack*$_2$
Suppose that

$$\frac{e\ \texttt{val}}{\texttt{unpack}[\alpha, x] = (\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau)\,\texttt{in}\,e_2 \mapsto [\tau'/\alpha][e/x]e_2}\ Eunpack_2$$

By Lemma 12.2 we get that $e^\circ\ \texttt{val}$. By rule *Eunpack*$_2$ we have that

$$\texttt{unpack}[\alpha, x] = \texttt{pack}[\tau'^\circ, e^\circ]\,\texttt{as}\,\exists\alpha.\tau^\circ\,\texttt{in}\,e_2^\circ \mapsto [\tau'^\circ/\alpha][e^\circ/x]e_2^\circ$$

Since $(\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau)^\circ = \texttt{pack}[\tau'^\circ, e^\circ]\,\texttt{as}\,\exists\alpha.\tau^\circ$, this is equivalent to

$$\texttt{unpack}[\alpha, x] = (\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau)^\circ\,\texttt{in}\,e_2^\circ \mapsto [\tau'^\circ/\alpha][e^\circ/x]e_2^\circ$$

Which is also equivalent to $(\texttt{unpack}[\alpha, x] = (\texttt{pack}[\tau', e]\,\texttt{as}\,\exists\alpha.\tau)\,\texttt{in}\,e_2)^\circ \mapsto ([\tau'/\alpha][e/x]e_2)^\circ$.

□

Corollary 12.4. If $\cdot \vdash_C e : \tau$ and $e^\circ \downarrow$, then $e \downarrow$.

Proof. Suppose $\cdot \vdash_C e : \tau$ and $e^\circ \downarrow$. Suppose e does not halt. But by Lemma 12.3 this would mean that e° does not halt, as every time e can take a step, so can e°. But this is a contradiction, so $e \downarrow$. □

13 Equivalence Preservation

Theorem 13.1. If $\Delta; \Gamma \vdash_S e \cong e' : \tau$, then $\Delta; \Gamma \vdash_C e \cong e' : \tau$.

Proof. Suppose that $\Delta; \Gamma \vdash_S e \cong e' : \tau$. Let $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \mathtt{int})$ be some context in the combined language. We want to show that $\mathcal{C}\{e\} \simeq \mathcal{C}\{e'\}$. Suppose $\mathcal{C}\{e\} \downarrow$, which means that there is some $v\,\mathtt{val}$ such that $\mathcal{C}\{e\} \mapsto^* v$. By Lemma 12.3, we get that $(\mathcal{C}\{e\})^\circ \mapsto^* v^\circ$, and by Lemma 12.2 we know that since $v\,\mathtt{val}$ it must be that $v^\circ\,\mathtt{val}$. Thus $(\mathcal{C}\{e\})^\circ \downarrow$ as well.

Note that since $\Delta; \Gamma \vdash_S e \cong e' : \tau$ is in the source language, we know that $\Gamma^\circ = \Gamma$, $\tau^\circ = \tau$, $e^\circ = e$, and $e'^\circ = e'$. This implies that $\mathcal{C}^\circ : (\Delta; \Gamma^\circ \triangleright \tau^\circ) \rightsquigarrow (\cdot \triangleright \mathtt{int})$ can equivalently be typed as $\mathcal{C}^\circ : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \mathtt{int})$. This is a context in the source language, and since $\Delta; \Gamma \vdash_S e \cong e' : \tau$ by assumption, we know that by definition $\mathcal{C}^\circ\{e\} \simeq \mathcal{C}^\circ\{e'\}$. Since we know that $(\mathcal{C}\{e\})^\circ = \mathcal{C}^\circ\{e^\circ\} = \mathcal{C}^\circ\{e\} \downarrow$, this implies that $\mathcal{C}^\circ\{e'\} \downarrow$. By the above this is equivalent to saying that $(\mathcal{C}\{e'\})^\circ \downarrow$. Then by Corollary 12.4, we have that $\mathcal{C}\{e'\} \downarrow$.

Therefore, $\mathcal{C}\{e\} \downarrow \Rightarrow \mathcal{C}\{e'\} \downarrow$. Showing the reverse can be proved in a symmetric way. This gives us that $\mathcal{C}\{e\} \simeq \mathcal{C}\{e'\}$, so therefore $\Delta; \Gamma \vdash_C e \cong e' : \tau$, as desired. ☐

Lemma 13.2. If $\Delta; \Gamma \vdash_C e \cong e' : \tau$, $\Delta; \Gamma \vdash_T e : \tau$, and $\Delta; \Gamma \vdash_T e' : \tau$, then $\Delta; \Gamma \vdash_T e \cong e' : \tau$.

Proof. Suppose that $\Delta; \Gamma \vdash_S e \cong e' : \tau$, $\Delta; \Gamma \vdash_T e : \tau$, and $\Delta; \Gamma \vdash_T e' : \tau$. Let $\mathcal{C} : (\Delta; \Gamma \triangleright \tau) \rightsquigarrow (\cdot \triangleright \mathtt{int})$ be some context in the target language. We want to show that $\mathcal{C}\{e\} \simeq \mathcal{C}\{e'\}$. However this follows immediately from our assumption that $\Delta; \Gamma \vdash_S e \cong e' : \tau$, as a context in the target language is also in the combined language. ☐

Theorem 13.3. If $\Delta; \Gamma \vdash_S e_1 \cong e_2 : \tau$, $\Delta; \Gamma \vdash_S e_1 : \tau \rightsquigarrow \bar{e_1}$, and $\Delta; \Gamma \vdash_S e_2 : \tau \rightsquigarrow \bar{e_2}$, then $\Delta; |\Gamma| \vdash_T \bar{e_1} \cong \bar{e_2} : |\tau|$.

Proof. Suppose that $\Delta; \Gamma \vdash_S e_1 \cong e_2 : \tau$, $\Delta; \Gamma \vdash_S e_1 : \tau \rightsquigarrow \bar{e_1}$, and $\Delta; \Gamma \vdash_S e_2 : \tau \rightsquigarrow \bar{e_2}$. By Theorem 13.1, we know that $\Delta; \Gamma \vdash_C e_1 \cong e_2 : \tau$. By Theorem 11.5 we get that

$$\Delta; \Gamma \vdash_C e_1 \cong \mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e_1}) : \tau$$

$$\Delta; \Gamma \vdash_C e_2 \cong \mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e_2}) : \tau$$

Thus by transitivity we have that

$$\Delta; \Gamma \vdash_C \mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e_1}) \cong \mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e_2}) : \tau$$

By the congruence of contextual equivalence, we get that

$$\Delta; \Gamma \vdash_C \mathtt{over}_\tau(\mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e_1})) \cong \mathtt{over}_\tau(\mathtt{back}_\tau([\mathtt{over}_\Gamma/\Gamma]\bar{e_2})) : |\tau|$$

Then by Lemma 11.1 we have that

$$\Delta; \Gamma \vdash_C [\mathtt{over}_\Gamma/\Gamma]\bar{e_1} \cong [\mathtt{over}_\Gamma/\Gamma]\bar{e_2} : |\tau|$$

Again by the congurence of contextual equivalence we have that

$$\Delta; |\Gamma| \vdash_C [\mathtt{back}_\Gamma/\Gamma][\mathtt{over}_\Gamma/\Gamma]\bar{e_1} \cong [\mathtt{back}_\Gamma/\Gamma][\mathtt{over}_\Gamma/\Gamma]\bar{e_2} : |\tau|$$

Then again by Lemma 11.1, we get that $\Delta; |\Gamma| \vdash_C \bar{e_1} \cong \bar{e_2} : |\tau|$, and since both $\bar{e_1}$ and $\bar{e_2}$ are in the target language, we have by Lemma 13.2 that $\Delta; |\Gamma| \vdash_T \bar{e_1} \cong \bar{e_2} : |\tau|$. ☐