Syntax and Semantics of Cedille

Aaron Stump, Chris Jenkins
Computer Science
The University of Iowa
aaron-stump@uiowa.edu

1 Introduction

The type theory of Cedille is called the Calculus of Dependent Lambda Eliminations (CDLE). This document presents the version of CDLE as of December 1, 2020. We have made many changes from the first paper on CDLE [15], mostly in the form of dropping constructs we discovered (to our surprise) could be derived [16]. We have also omitted *lifting* – a technique for large eliminations with lambda encodings – in this document's version of CDLE. Some uses of lifting can be simulated other ways within the system, though the limits of this are still under investigation. We also include a construct δ, for deriving a contradiction from a proof that lambda-encoded true equals lambda-encoded false. This also compensates somewhat for the lack of lifting.

At a high level, CDLE is an extrinsic (i.e., Curry-style) type theory extending the Calculus of Constructions with three additional constructs, which allow deriving induction principles within the theory for inductive datatypes. The goal is to support usual idioms of dependently typed programming and proving as in Agda or similar tools, but using pure lambda encodings for all data, and requiring a much smaller core theory.

The current Cedille implementation of CDLE extends the system described below with a number of features intended to make programming in the system more convenient and with less redundancy. These features all compile away to a slightly simplified version of the theory presented in this document, called Cedille Core, described here: `https://github.com/astump/cedille-core-spec`.

2 Classification Rules

The classification rules are given in Figures 1, 2, and 3, with Figure 4 giving the context formation rules. For brevity, we take these figures as implicitly specifying the syntax of contexts Γ, kinds κ, types T, and annotated terms t; these may use term variables x and type variables X, which we assume come from distinct sets. So terms and types are syntactically distinguished. We follow the syntax of our implementation Cedille, which distinguishes application of a term or type e to a type ($e \cdot T$), from application to a term ($e\ t$), and application to an erased term argument (e -t, in which case e must be a term). Note that center dot ($\cdot$) is also used to denote the empty type context; since the usage for typing contexts always has the symbol occur to the left of the turnstile ($\vdash$), no confusion should arise from overloading notation.

The typing rules (Figure 3) are bidirectional [14], while the kinding and superkinding rules (Figure 2 and 1) are unidirectional (i.e., synthesizing only). We write $\Leftrightarrow$ to range over $\{\Leftarrow, \Rightarrow\}$, and when this symbol occurs multiple times in a rule, it is intended that such occurrences be read the same way (i.e., read the occurrences as either all $\Leftarrow$ or all $\Rightarrow$). The rules are intended to be read bottom-up as an algorithm (in a standard way, c.f. [12, 13]) for synthesizing a classifier from a context and an expression (*type synthesis*, $\Rightarrow$) or checking an expression against a classifier in a context (*type checking*, $\Leftarrow$). Since variables have their type synthesized, and since we sometimes wish to substitute a variable with a term whose type can only be checked, we define

$$\frac{}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash \kappa}{\Gamma \vdash \Pi\, x{:}T.\, \kappa} \qquad \frac{\Gamma \vdash \kappa' \quad \Gamma, X : \kappa' \vdash \kappa}{\Gamma \vdash \Pi\, X{:}\kappa'.\, \kappa}$$

Figure 1: Rules for checking that a kind is well-formed ($\Gamma \vdash \kappa$)

$$[t/x]^T \;=\; [\chi\ T\ \text{-}\ t/x]$$

$$\frac{(X : \kappa) \in \Gamma}{\Gamma \vdash X \Rightarrow \kappa} \qquad \frac{\Gamma \vdash \kappa \quad \Gamma, X : \kappa \vdash T \Rightarrow \star}{\Gamma \vdash \forall\, X{:}\kappa.\, T \Rightarrow \star}$$

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \star}{\Gamma \vdash \forall\, x{:}T.\, T' \Rightarrow \star} \qquad \frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \star}{\Gamma \vdash \Pi\, x{:}T.\, T' \Rightarrow \star}$$

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \kappa}{\Gamma \vdash \lambda\, x{:}T.\, T' \Rightarrow \Pi\, x{:}T.\, \kappa} \qquad \frac{\Gamma \vdash \kappa \quad \Gamma, X : \kappa \vdash T' \Rightarrow \kappa'}{\Gamma \vdash \lambda\, X{:}\kappa.\, T' \Rightarrow \Pi\, X{:}\kappa.\, \kappa'}$$

$$\frac{\Gamma \vdash T \Rightarrow \Pi\, x{:}T'.\, \kappa \quad \Gamma \vdash t \Leftarrow T'}{\Gamma \vdash T\ t \Rightarrow [t/x]^{T'} \kappa} \qquad \frac{\Gamma \vdash T_1 \Rightarrow \Pi\, X{:}\kappa_2.\, \kappa_1 \quad \Gamma \vdash T_2 \Rightarrow \kappa_2' \quad \kappa_2 \cong \kappa_2'}{\Gamma \vdash T_1 \cdot T_2 \Rightarrow [T_2/X]\kappa_1}$$

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \star}{\Gamma \vdash \iota\, x{:}T.\, T' \Rightarrow \star} \qquad \frac{FV(t\ t') \subseteq dom(\Gamma)}{\Gamma \vdash \{t \simeq t'\} \Rightarrow \star}$$

Figure 2: Rules for synthesizing a kind for a type ($\Gamma \vdash T \Rightarrow \kappa$)

a shorthand notation: $[t/x]^T$ means $[\chi\ T\ \text{-}\ t/x]$, where χ is the construct for explicit type annotations (see Section 2.1).

In terms, when the type of an introduction form is checked or the type of an elimination form is synthesized, we use call-by-name normalization for types $\rightsquigarrow_\beta^*$ to put types in weak head normal form, revealing type constructors. We abbreviate the conjuction "term t synthesizes some type, and that type call-by-name reduces to another type" with the symbol $\overset{\rightsquigarrow_\beta^*}{\Rightarrow}$, defined formally at the top of Figure 3. When a redex (reducible expression) occurs such that the argument to a type is a term, such as $(\lambda\, x : T_1.\, T_2)\ t_1$, the reduction uses substitution with annotations to $[t_1/x]^{T_1}\ T_2$.

Call-by-name reduction is *not* what underpins the congruence relation $\cong$ for types or terms, which is full β-equivalence (for types) and $\beta\eta$-equivalence (for terms) modulo erasure of annotations in terms. The erasure operation is defined in Figure 5, and is essentially the *extraction* function for the Implicit Calculus of Constructions given by [1] adapted to CDLE. To understand the role of erasure, recall that the type theory is *extrinsic* (a.k.a. Curry-style), and hence we only consider erasures $|t|$ of terms when testing convertibility. This is lifted to the conversion relation on types $T \cong T'$ and kinds $\kappa \cong \kappa'$, whose central rules are given in Figure 6. That figure omits the various congruence rules needed to equate bigger expressions by equating subexpressions (the convertibility rules for kinds consists entirely of congruence rules).

2.1 Overview of the constructs

CDLE has as a subsystem the extrinsic Calculus of Constructions (CC). We have dependent types $\Pi\, x{:}T.\, T'$ and kinds $\Pi\, x{:}T.\, \kappa$, as well as term- and type-level quantification over (possibly higher-kinded) types $\forall\, X{:}\kappa.\, T$ and $\Pi\, X{:}\kappa.\, \kappa'$. We use $\forall$ when the corresponding argument will be erased, and Π when it will be retained. Since we do not erase term or type arguments from type-level applications, we thus write $\Pi\, X{:}\kappa.\, \kappa'$ instead

$$\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} T \;=\; \exists T'.\, \Gamma \vdash t \Rightarrow T' \wedge T' \rightsquigarrow_\beta^* T$$

$$\frac{(x : T) \in \Gamma}{\Gamma \vdash x \Rightarrow T} \qquad \frac{\Gamma \vdash t \Rightarrow T' \quad T' \cong T}{\Gamma \vdash t \Leftarrow T}$$

$$\frac{T \rightsquigarrow_\beta^* \Pi\, x{:}T_1.\, T_2 \quad \Gamma, x : T_1 \vdash t \Leftarrow T_2}{\Gamma \vdash \lambda\, x.\, t \Leftarrow T} \qquad \frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \Pi\, x{:}T'.\, T \quad \Gamma \vdash t' \Leftarrow T'}{\Gamma \vdash t\ t' \Rightarrow [t'/x]^{T'} T}$$

$$\frac{T' \rightsquigarrow_\beta^* \forall\, X{:}\kappa.\, T \quad \Gamma, X : \kappa \vdash t \Leftarrow T}{\Gamma \vdash \Lambda\, X.\, t \Leftarrow T'} \qquad \frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \forall\, X{:}\kappa.\, T \quad \Gamma \vdash T' \Rightarrow \kappa' \quad \kappa' \cong \kappa}{\Gamma \vdash t \cdot T' \Rightarrow [T'/X]T}$$

$$\frac{T \rightsquigarrow_\beta^* \forall\, x{:}T_1.\, T_2 \quad \Gamma, x : T_1 \vdash t \Leftarrow T_2 \quad x \notin FV(|t|)}{\Gamma \vdash \Lambda\, x.\, t \Leftarrow T} \qquad \frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \forall\, x{:}T'.\, T \quad \Gamma \vdash t' \Leftarrow T'}{\Gamma \vdash t \text{ -}t' \Rightarrow [t'/x]^{T'} T}$$

$$\frac{\begin{array}{c} T \rightsquigarrow_\beta^* \iota\, x{:}T_1.\, T_1 \qquad \Gamma \vdash t_1 \Leftarrow T_1 \\ \Gamma \vdash t_2 \Leftarrow [t/x]^{T_1}\ T_2 \quad |t_1| =_{\beta\eta} |t_2| \end{array}}{\Gamma \vdash [t_1, t_2] \Leftarrow T} \qquad \frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \iota\, x{:}T.\, T'}{\Gamma \vdash t.1 \Rightarrow T}$$

$$\frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \iota\, x{:}T.\, T'}{\Gamma \vdash t.2 \Rightarrow [t.1/x]T'} \qquad \frac{T \rightsquigarrow_\beta^* \{t_1 \simeq t_2\} \quad FV(t') \subseteq \mathit{dom}(\Gamma) \quad |t_1| =_{\beta\eta} |t_2|}{\Gamma \vdash \beta\{t'\} \Leftarrow T}$$

$$\frac{\Gamma \vdash t \Leftarrow \{\lambda\, x.\, \lambda\, y.\, x \simeq \lambda\, x.\, \lambda\, y.\, y\}}{\Gamma \vdash \delta \text{ - } t \Leftarrow T} \qquad \frac{\begin{array}{c} \Gamma \vdash t' \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \{t_1 \simeq t_2'\} \quad FV(t_2) \subseteq \mathit{dom}(\Gamma) \quad |t_2'| =_{\beta\eta} |t_2| \\ \Gamma \vdash [t_2/x]\ T \Rightarrow \star \qquad \Gamma \vdash t \Rightarrow T' \qquad T' \cong [t_1/x]T \end{array}}{\Gamma \vdash \rho\ t'\ @x\langle t_2\rangle.T \text{ - } t \Rightarrow [t_2/x]\ T}$$

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma \vdash t \Leftarrow T}{\Gamma \vdash \chi\ T \text{ - } t\ \Rightarrow T} \qquad \frac{\begin{array}{c} \Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \{t_1 \simeq t_2\} \qquad\qquad \Gamma \vdash t' \Leftrightarrow T \\ |t_1| =_{\beta\eta} |t'| \qquad |t_2| =_{\beta\eta} |t''| \quad FV(t'') \subseteq \mathit{dom}(\Gamma) \end{array}}{\Gamma \vdash \varphi\ t \text{ - } t'\ \{t''\} \Leftrightarrow T}$$

Figure 3: Rules for checking a term against a type ($\Gamma \vdash t \Leftarrow T$) and synthesizing a type for a term ($\Gamma \vdash t \Rightarrow T$)

$$\frac{}{\vdash \cdot} \qquad \frac{\Gamma \vdash T \Rightarrow \star}{\vdash \Gamma, x : T} \qquad \frac{\Gamma \vdash \kappa}{\vdash \Gamma, X : \kappa}$$

Figure 4: Rules for checking a context is well-formed

$$
\begin{array}{lcllcl}
|x| & = & x & |\lambda\, x.\, t| & = & \lambda\, x.\, |t| \\
|t\ t'| & = & |t|\ |t'| & |t \cdot T| & = & |t| \\
|\Lambda\, x.\, t| & = & |t| & |t\ \text{-}t'| & = & |t| \\
|[t, t']| & = & |t| & |t.1| & = & |t| \\
|t.2| & = & |t| & |\beta\{t\}| & = & |t| \\
|\delta\ \text{-}\ t| & = & \lambda\, x.\, x & |\rho\ t\ @x.T\ \text{-}\ t'| & = & |t'| \\
|\varphi\ t\ \text{-}\ t'\ \{t''\}| & = & |t''| & |\chi\ T\ \text{-}\ t'| & = & |t'|
\end{array}
$$

Figure 5: Erasure for annotated terms

$$
\frac{T \rightsquigarrow_\beta^* T_1 \quad T' \rightsquigarrow_\beta^* T_2 \quad T_1 \cong^t T_2}{T \cong T'}
\qquad
\frac{T \cong^t T'}{T \cong T'}
$$

$$
\frac{T \cong^t T' \quad |t| =_{\beta\eta} |t'|}{T\ t \cong^t T\ t'}
\qquad
\frac{|t_1| =_{\beta\eta} |t_1'| \quad |t_2| =_{\beta\eta} |t_2'|}{\{t_1 \simeq t_2\} \cong^t \{t_1' \simeq t_2'\}}
$$

Figure 6: Non-congruence rules for conversion

of $\forall\, X : \kappa.\, \kappa'$. For abstractions, we write λ to correspond to Π and Λ to correspond to $\forall$. As noted above, application to a type is denoted with center dot $(\cdot)$.

To Curry-style CC, CDLE adds: implicit products, introduced orginially by Miquel [10]; a primitive equality type $\{t \simeq t'\}$; and dependent intersection types $\iota\, x\!:\!T.\, T'$, introduced by Kopylov [9]. Implicit products are used for erased arguments to functions, found also in systems like Agda (c.f. [11]). Dependent intersections are a rather exotic construct allowing us to assign type $\iota\, x\!:\!T_1.\, T_2$ to erased term t when we can assign T_1 to t, and also assign $[t/x]T$ to t. For an annotated introduction form, we write $[t_1, t_2]$, where t_1 checks against type T_1, t_2 checks against $[t_1/x]^{T_1}T_2$, and t_1 and t_2 have $\beta\eta$-equivalent erasures. Dependent intersections thus enable a controlled form of self-reference in the type. Previous work showed how to use this to derive induction for Church-encoded natural numbers [16]. We will see below further uses of this construct.

The typing rules include conversion checks in a few places, e.g., as is standard when switching from checking to synthesizing mode. As mentioned earlier, two rules near the top of Figure 3 state that one may (non-deterministically) call-by-name reduce the type one is synthesizing or checking, before proceeding. We also include the construct $\chi\ T\ \text{-}\ t$ which allows a term whose type can be checked to be ascribed a user-provided type so that the whole expression may occur where a term whose type can be synthesized is expected.

We have modified the rules for equality types $\{t_1 \simeq t_2\}$ so that we require nothing of t_1 and t_2 except that the set $dom(\Gamma)$ of variables declared by Γ includes their free variables $FV(t_1\ t_2)$. Further modifications over the version of CDLE in [16] are:

- To prove $\{t_1 \simeq t_2\}$ for definitionally equal terms (that is, terms that are $\beta\eta$-equivalent modulo erasure), one now writes $\beta\{t'\}$, with the critical idea that $|\beta\{t'\}|$ erases to (the possibly unrelated) $|t'|$. We call this the **Kleene trick**, because it goes back to Kleene's numeric realizability [8], which accepts any number n as a realizer of a true equation. Here, we accept any term t' as a realizer of $\{t_1 \simeq t_2\}$, provided the free variables of t' are declared in the context.

  The Kleene trick means that in Cedille, any such term — even otherwise untypable terms, non-normalizing terms, etc. — prove trivially true equations. Put another way, any trivially true equation type in CDLE is a suitable type to classify all untyped lambda calculus terms.

- The ρ construct allows one to rewrite occurrences of t_1 to t_2' in the synthesized type of its second subexpression. The version presented here requires a type annotations ("guide") $@x\langle t_2\rangle.T$, where x indicates the occurrences of t_1 to substitute and t_2 is a term, $\beta\eta$-equivalent to t_2', that is used to replace

x. There is no requirement that t_1 and t_2 will have the same type, or even be typable. However, $[t_2/x]\,T$ must be a well-kinded type and $[t_1/x]T$ (which might be ill-kinded) must be convertible with T' modulo erasure. In Cedille, the guide is optional and the construct may be used to rewrite a contextually given type; a heuristic, whose details are beyond the scope of this document, is used to produce a resulting type that is well-kinded.

- We adopt a strong form of Nuprl's **direct computation rules** [3]: If we have a term t' of type T and a proof t that $\{t' \simeq t''\}$, then we may conclude that t'' has type T by writing the annotated term $\varphi\ t$ - t' $\{t''\}$, which erases to t''.
- Where the previous version of CDLE uses β-equivalence for (erased) terms, we here adopt $\beta\eta$-equivalence. This allows us to observe in many cases that retyping functions are actually $\beta\eta$-equivalent to $\lambda\, x.\, x$. While $\beta\eta$-equivalence takes more work to incorporate into intrinsic type theory [6], it raises no difficulties for our extrinsic one.
- In this version, we add an explicit axiom δ saying that Church-encoded boolean *true* is different from *false*. In the implementation of Cedille, this is generalized to the rule where the proof t synthesizes an equation $\{t_1 \simeq t_2\}$ in which $|t_1|$ and $|t_2|$ are separable using the *Böhm-out algorithm* [2].

  In the first version of CDLE, such an axiom was derivable from *lifting*, a construct allowing terms with simple types to be lifted to the type level [15]. We omit lifting in this new version of CDLE, because while sound, lifting as defined in that previous work is complicated and appears to be incomplete. Developing a new form of lifting remains to future work.

The equality type remains **intensional**: we equate terms iff they are $\beta\eta$-equal.

2.2 Semantics and metatheory

Figure 7 gives a realizability semantics for types and kinds, following the semantics given in the previous papers on CDLE [16, 15]. Details of this semantics are presented further in Section 2.3 below. Using the semantics and the definition in Figure 8 of $[\![\Gamma]\!]$, we can prove the following theorem:

Theorem 1 (Soundness). *Suppose $(\sigma, \rho) \in [\![\Gamma]\!]$. Then we have:*

1. *If $\Gamma \vdash \kappa$, then $[\![\kappa]\!]_{\sigma,\rho}$ is defined.*
2. *If $\Gamma \vdash T \Rightarrow \kappa$, then $[\![T]\!]_{\sigma,\rho} \in [\![\kappa]\!]_{\sigma,\rho}$.*
3. *If $\Gamma \vdash t \Rightarrow T$ then $[\sigma|t|]_{c\beta\eta} \in [\![T]\!]_{\sigma,\rho} \in \mathcal{R}$.*
4. *If $\Gamma \vdash t \Leftarrow T$ and $[\![T]\!]_{\sigma,\rho} \in \mathcal{R}$, then $[\sigma|t|]_{c\beta\eta} \in [\![T]\!]_{\sigma,\rho} \in \mathcal{R}$.*
5. *If $T \cong T'$ or $T \cong^t T'$ and $[\![T]\!]_{\sigma,\rho}$ and $[\![T']\!]_{\sigma,\rho}$ are both defined, then they are equal.*
6. *If $\kappa \cong \kappa'$ and $[\![\kappa]\!]_{\sigma,\rho}$ and $[\![\kappa']\!]_{\sigma,\rho}$ are both defined, then they are equal.*

An easy corollary, by the semantics of $\forall$-types, is then:

Theorem 2 (Logical consistency). *There is no term t such that $\cdot \vdash t : \forall\, X : \star.\, X$.*

It may worry some readers that we have:

Observation 3. *There are typable terms t which fail to normalize.*

Defining `Top` to be $\{\lambda\, x.\, x \simeq \lambda\, x.\, x\}$, we may assign `Top` to any closed term `t`, including non-normalizing ones. In our annotated syntax, we write $\beta\{$`t`$\}$. Even without this, the presence of φ allows us to type non-normalizing terms assuming an erased argument x of type $\{\lambda\, x.\, x \simeq \lambda\, x.\, x\ x\}$ by changing the type of the term `id` $\cdot$ `True id`, where `True` is $\forall\, X : \star.\, X \to X$. This would allow us to give the type `True` to $\Omega = (\lambda\, x.\, x\ x)\ \lambda\, x.\, x\ x$. In general, we can use any inconsistent assumption to do this, and in the presence of δ that includes all equations between two terms that are Böhm-separable. But, failure of normalization does not impinge on Theorem 2. Extensional Martin-Löf type theory (MLTT) is also non-normalizing, for a very similar reason, but this fact does not contradict its logical soundness [5]. In CDLE, the guarantees one gets about the behavior of terms are expressed almost entirely in their types. If the types are weak, then not much is guaranteed; but stronger types can guarantee properties like normalization, as demonstrated by the following theorem:

Theorem 4 (Call-by-name normalization of functions). *Suppose $\cdot \vdash t \Rightarrow T$ and there exists a closed term t' such that $\cdot \vdash t' \Rightarrow T \to \Pi\, x\!:\!T_1.\, T_2$ for some T_1, T_2 and $|t'| = \lambda\, x.\, x$. Then $|t|$ is call-by-name normalizing.*

Given the lack of normalization in general, several checks in the typing rules – for things like $|t| =_{\beta\eta} |t'|$ – are formally undecidable. We simply impose a bound on the number of steps of reduction, and thus restore formal decidability (we are checking "typable within a given budget"). In practice, the same is done for Coq and Agda, where type checking is decidable but, in general, infeasible (since one may write astronomically slow terminating functions).

Finally, in line with ideas recently advocated by Dreyer, we are less concerned with syntactic type preservation as we are with *semantic* type preservation [4]. Note that by construction, semantic types $[\![T]\!]_{\sigma,\rho}$ are preserved by $\beta\eta$-reduction:

Theorem 5 (Semantic type preservation). *If $t \leadsto_{\beta\eta} t'$ and $t \in [\![T]\!]_{\sigma,\rho}$, then $t' \in [\![T]\!]_{\sigma,\rho}$.*

Confluence of $\beta\eta$-reduction for (erased) terms is nothing other than confluence of untyped lambda calculus. This is because, as easily verified by inspecting Figure 5, the erasure function maps annotated terms t to terms $|t|$ of pure untyped lambda calculus.

We make a concession to syntactic classifier preservation in the case of types and kinds. During type inference, types may be reduced using a call-by-name operational semantics to reveal type constructors. With the removal of lifting from CDLE, terms cannot compute types and so no terms need to be reduced during this process.

Theorem 6 (Syntactic kind preservation). *If $\Gamma \vdash T \Rightarrow \kappa$ and $T \leadsto_{\beta} T'$ then $\Gamma \vdash T' \Rightarrow \kappa'$ for some κ' such that $\kappa \cong \kappa'$.*

From this theorem and a few other lemmas (see Appendix C), we can show the *validity* (or *agreement*) of the judgments comprising CDLE.

Theorem 7 (Judgment validity). *If $\vdash \Gamma$ then:*

1. *if $\Gamma \vdash T \Rightarrow \kappa$ then $\Gamma \vdash \kappa$*
2. *if $\Gamma \vdash t \Rightarrow T$ then $\Gamma \vdash T \Rightarrow \star$*

2.3 Some details about the semantics and the proof of Theorem 1

Following the development in [15], we work with set-theoretic partial functions for the semantics of higher-kinded types. Types are interpreted as $\beta\eta$-closed sets of closed terms. Let $\mathcal{L}$ be the set of closed terms of pure lambda calculus (differently from [15], we include all terms at this point, even non-normalizing ones). We write $=_{c\beta\eta}$ for standard $\beta\eta$-equivalence of pure lambda calculus, restricted to closed terms; and $[t]_{c\beta\eta}$ for $\{t' \mid t =_{c\beta\eta} t'\}$. This is extended to sets S of terms by writing $[S]_{c\beta\eta}$ for $\{[t]_{c\beta\eta} \mid t \in S\}$. If (in our meta-language) we affirm a statement involving application of a partial function, then it is to be understood that that application is defined.

Definition 8 (Reducibility candidates). $\mathcal{R} := \{[S]_{c\beta\eta} \mid S \subseteq \mathcal{L}\}$.

Throughout the development we find it convenient to use a **choice function** ζ. Given any nonempty set E of terms, ζ returns some element of E. Note that if $a \in A \in \mathcal{R}$, then a is a nonempty set of terms of pure lambda calculus; it can also happen that $A \in \mathcal{R}$ is empty. The proof of Theorem 1 (see appendix) is then a straightforward adaptation of [15].

$$
\begin{array}{lcl}
\llbracket X \rrbracket_{\sigma,\rho} & = & \rho(X) \\
\llbracket \Pi x : T_1.T_2 \rrbracket_{\sigma,\rho} & = & [\{\lambda x.t \mid \forall E \in \llbracket T_1 \rrbracket_{\sigma,\rho}. \\
 & & \quad [[\zeta(E)/x]t]_{c\beta\eta} \in \llbracket T_2 \rrbracket_{\sigma[x \mapsto \zeta(E)],\rho} \ \wedge \ t = |t|\}]_{c\beta\eta} \\
\llbracket \forall X : \kappa.T \rrbracket_{\sigma,\rho} & = & \cap\{\llbracket T \rrbracket_{\sigma,\rho[X \mapsto S]} \mid S \in \llbracket \kappa \rrbracket_{\sigma,\rho}\} \\
\llbracket \forall x : T.T' \rrbracket_{\sigma,\rho} & = & \cap_\star\{\llbracket T' \rrbracket_{\sigma[x \mapsto \zeta(E)],\rho} \mid E \in \llbracket T \rrbracket_{\sigma,\rho}\} \\
\llbracket \iota x : T.T' \rrbracket_{\sigma,\rho} & = & \{E \in \llbracket T \rrbracket_{\sigma,\rho} \mid E \in \llbracket T' \rrbracket_{\sigma[x \mapsto \zeta(E)],\rho}\} \\
\llbracket \lambda X : \kappa.T \rrbracket_{\sigma,\rho} & = & (S \in \llbracket \kappa \rrbracket_{\sigma,\rho} \mapsto \llbracket T \rrbracket_{\sigma,\rho[X \mapsto S]}) \\
\llbracket \lambda x : T.T' \rrbracket_{\sigma,\rho} & = & (E \in \llbracket T \rrbracket_{\sigma,\rho} \mapsto \llbracket T' \rrbracket_{\sigma[x \mapsto \zeta(E)],\rho}) \\
\llbracket T\ T' \rrbracket_{\sigma,\rho} & = & \llbracket T \rrbracket_{\sigma,\rho}(\llbracket T' \rrbracket_{\sigma,\rho}) \\
\llbracket T\ t \rrbracket_{\sigma,\rho} & = & \llbracket T \rrbracket_{\sigma,\rho}([\sigma|t|]_{c\beta\eta}) \\
\llbracket \{t \simeq t'\} \rrbracket_{\sigma,\rho} & = & [\{t'' \mid \sigma|t| =_{\beta\eta} \sigma|t'| \ \wedge \ t'' = |t''|\}]_{c\beta\eta} \\
 & & \quad \text{if } FV(t\ t') \subseteq dom(\sigma) \\
\llbracket \star \rrbracket_{\sigma,\rho} & = & \mathcal{R} \\
\llbracket \Pi x : T.\kappa \rrbracket_{\sigma,\rho} & = & (E \in \llbracket T \rrbracket_{\sigma,\rho} \to \llbracket \kappa \rrbracket_{\sigma[x \mapsto \zeta(E)],\rho}), \\
 & & \quad \text{if } \llbracket T \rrbracket_{\sigma,\rho} \in \mathcal{R} \\
\llbracket \Pi x : \kappa.\kappa' \rrbracket_{\sigma,\rho} & = & (S \in \llbracket \kappa \rrbracket_{\sigma,\rho} \to \llbracket \kappa \rrbracket_{\sigma,\rho[X \mapsto S]}) \\
\cap_\star X & = & \left\{ \begin{array}{l} \cap X, \ \text{if } X \neq \emptyset \\ [\mathcal{L}]_{c\beta\eta}, \ \text{otherwise} \end{array} \right.
\end{array}
$$

Figure 7: Semantics for types and kinds

$$
\begin{array}{lcl}
(\sigma \uplus [x \mapsto t], \rho) \in \llbracket \Gamma, x : T \rrbracket & \Leftrightarrow & (\sigma, \rho) \in \llbracket \Gamma \rrbracket \ \wedge \ [t]_{c\beta\eta} \in \llbracket T \rrbracket_{\sigma,\rho} \in \mathcal{R} \ \wedge \ t = |t| \\
(\sigma, \rho \uplus [X \mapsto S]) \in \llbracket \Gamma, X : \kappa \rrbracket & \Leftrightarrow & (\sigma, \rho) \in \llbracket \Gamma \rrbracket \ \wedge \ S \in \llbracket \kappa \rrbracket_{\sigma,\rho} \\
(\emptyset, \emptyset) \in \llbracket \cdot \rrbracket & &
\end{array}
$$

Figure 8: Semantics of typing contexts Γ

Acknowledgments. This work was partially supported by the US NSF support under award 1524519, and US DoD support under award FA9550-16-1-0082 (MURI program).

References

[1] Bruno Barras and Bruno Bernardo. The implicit calculus of constructions as a programming language with dependent types. In Roberto M. Amadio, editor, *Foundations of Software Science and Computational Structures, 11th International Conference, FOSSACS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29 - April 6, 2008. Proceedings*, volume 4962 of *Lecture Notes in Computer Science*, pages 365–379. Springer, 2008.

[2] C. Böhm, M. Dezani-Ciancaglini, P. Peretti, and S.Ronchi Della Rocca. A discrimination algorithm inside lambda-beta-calculus. *Theoretical Computer Science*, 8(3):271 − 291, 1979.

[3] Robert L. Constable, Stuart F. Allen, Mark Bromley, Rance Cleaveland, J. F. Cremer, R. W. Harper, Douglas J. Howe, Todd B. Knoblock, N. P. Mendler, Prakash Panangaden, James T. Sasaki, and Scott F. Smith. *Implementing mathematics with the Nuprl proof development system.* Prentice Hall, 1986.

[4] Derek Dreyer. The Type Soundness Theorem That You Really Want to Prove (and Now You Can). Milner Award Lecture, delivered at Principles of Programming Languages (POPL), 2018.

[5] Peter Dybjer and Erik Palmgren. Intuitionistic Type Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University, winter 2016 edition, 2016.

[6] Herman Geuvers. The Church-Rosser Property for beta-eta-reduction in Typed lambda-Calculi. In *Proceedings of the Seventh Annual Symposium on Logic in Computer Science (LICS '92), Santa Cruz, California, USA, June 22-25, 1992*, pages 453–460. IEEE Computer Society, 1992.

[7] Ryo Kashima. A Proof of the Standardization Theorem in λ-Calculus. 2000. available from author's web page.

[8] S.C. Kleene. Classical Extensions of Intuitionistic Mathematics. In Y. Bar-Hillel, editor, *LMPS 2*, pages 31–44. North-Holland Publishing Company, 1965.

[9] Alexei Kopylov. Dependent intersection: A new way of defining records in type theory. In *18th IEEE Symposium on Logic in Computer Science (LICS)*, pages 86–95, 2003.

[10] Alexandre Miquel. The Implicit Calculus of Constructions Extending Pure Type Systems with an Intersection Type Binder and Subtyping. In Samson Abramsky, editor, *Typed Lambda Calculi and Applications*, volume 2044 of *Lecture Notes in Computer Science*, pages 344–359. Springer, 2001.

[11] Nathan Mishra-Linger and Tim Sheard. Erasure and Polymorphism in Pure Type Systems. In Roberto M. Amadio, editor, *Foundations of Software Science and Computational Structures, 11th International Conference, FOSSACS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29 - April 6, 2008. Proceedings*, volume 4962 of *Lecture Notes in Computer Science*, pages 350–364. Springer, 2008.

[12] Simon Peyton Jones, Dimitrios Vytiniotis, Stephanie Weirich, and Mark Shields. Practical Type Inference for Arbitrary-rank Types. *J. Funct. Program.*, 17(1):1–82, January 2007.

[13] Frank Pfenning. Lecture notes on bidirectional typing. https://www.cs.cmu.edu/ fp/courses/15312-f04/handouts/15-bidirectional.pdf, October 2001.

[14] Benjamin C. Pierce and David N. Turner. Local type inference. *ACM Trans. Program. Lang. Syst.*, 22(1):1–44, 2000.

[15] Aaron Stump. The Calculus of Dependent Lambda Eliminations. *J. Funct. Program.*, 27:e14, 2017.

[16] Aaron Stump. From Realizability to Induction via Dependent Intersection, 2018. in press.

A Proof of Theorem 1

First a few lemmas (easy proofs omitted):

Lemma 9. *$\llbracket\kappa\rrbracket_{\sigma,\rho}$ is nonempty if defined.*

Lemma 10. *If E is nonempty, then $[\zeta(E)]_{c\beta\eta} = E$*

Lemma 11. *The set $\mathcal{R}$ ordered by subset forms a complete lattice, with greatest element $[\mathcal{L}]_{c\beta\eta}$ and greatest lower bound of a nonempty set of elements given by intersection. Also, $\emptyset$ is the least element.*

Lemma 12 (Term substitution and interpretation)**.** *If $t' =_{c\beta\eta} \sigma|t|$, then:*

- $\llbracket T\rrbracket_{\sigma[x\mapsto t'],\rho} = \llbracket[t/x]T\rrbracket_{\sigma,\rho}$
- $\llbracket\kappa\rrbracket_{\sigma[x\mapsto t'],\rho} = \llbracket[t/x]\kappa\rrbracket_{\sigma,\rho}$

Note that Lemma 12 also applies to typed substitution: if $t' =_{c\beta\eta} \sigma|t|$ then by erasure so does $\chi\ T$ - t'.

Lemma 13 (Type substitution and interpretation)**.**

- $\llbracket T\rrbracket_{\sigma,\rho[X\mapsto\llbracket T'\rrbracket_{\sigma,\rho}]} = \llbracket[T'/X]T\rrbracket_{\sigma,\rho}$
- $\llbracket\kappa\rrbracket_{\sigma,\rho[X\mapsto\llbracket T'\rrbracket_{\sigma,\rho}]} = \llbracket[T'/X]\kappa\rrbracket_{\sigma,\rho}$

Lemma 14. *If $T \rightsquigarrow_{\beta}^{*} T'$ and $\llbracket T\rrbracket_{\sigma,\rho}$ is defined, then $\llbracket T'\rrbracket_{\sigma,\rho}$ is also defined and equals $\llbracket T\rrbracket_{\sigma,\rho}$.*

Proof. This follows by induction on the reduction derivation, making use of the previous substitution lemmas. □

Soundness (Theorem 1). The following proof is adapted from [15]. It proceeds by mutual induction on the assumed typing, kinding, or superkinding derivation, for each part of the lemma. We prove the parts successively.

A.1 Proof of part (1)

Case:

$$\frac{}{\Gamma \vdash \star}$$

$\llbracket\star\rrbracket_{\sigma,\rho}$ is just $\mathcal{R}$, which is defined.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash \kappa}{\Gamma \vdash \Pi\, x{:}T.\, \kappa}$$

By the IH, $\llbracket T\rrbracket_{\sigma,\rho} \in \mathcal{R}$, and so $\llbracket \Pi x : T.\, \kappa\rrbracket_{\sigma,\rho}$ is $(E \in \llbracket T\rrbracket_{\sigma,\rho} \to \llbracket\kappa\rrbracket_{\sigma[x\mapsto\zeta(E)],\rho})$. The latter quantity is defined if for all $E \in \llbracket T\rrbracket_{\sigma,\rho}$, $\llbracket\kappa\rrbracket_{\sigma[x\mapsto\zeta(E)],\rho})$ is, too. Since $\llbracket T\rrbracket_{\sigma,\rho} \in \mathcal{R}$, every element E of $\llbracket T\rrbracket_{\sigma,\rho}$ is nonempty, as noted above, so $\zeta(E)$ is defined. We may apply the IH to the second premise, since $(\sigma[x \mapsto \zeta(E)], \rho) \in \llbracket\Gamma, x : T\rrbracket$, because $E \in \llbracket T\rrbracket_{\sigma,\rho}$ (by assumption) and $[\zeta(E)]_{c\beta\eta} = E$. This gives definedness of the semantics of the Π-kind.

Case:

$$\frac{\Gamma \vdash \kappa' \quad \Gamma, X : \kappa' \vdash \kappa}{\Gamma \vdash \Pi\, X{:}\kappa'.\, \kappa}$$

We must show $(S \in [\![\kappa]\!]_{\sigma,\rho} \to [\![\kappa]\!]_{\sigma,\rho[X\mapsto S]})$ is defined. This is true if $[\![\kappa]\!]_{\sigma,\rho}$ is defined, which is the case by the IH applied to the first premise; and if for all $S \in [\![\kappa]\!]_{\sigma,\rho}$, $[\![\kappa]\!]_{\sigma,\rho[X\mapsto S]}$ is defined. The latter is true by the IH applied to the second premise.

A.2 Proof of part (2)

Case:

$$\frac{(X : \kappa) \in \Gamma}{\Gamma \vdash X \Rightarrow \kappa}$$

From the definition of $[\![\Gamma]\!]$, we obtain $\rho(X) \in [\![\kappa]\!]_{\sigma,\rho}$.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \star}{\Gamma \vdash \Pi\, x{:}T.\, T' \Rightarrow \star}$$

We must show $[\![\Pi x : T.T']\!]_{\sigma,\rho} \in \mathcal{R}$. The semantics defines $[\![\Pi x : T.T']\!]_{\sigma,\rho}$ to be $[A]_{c\beta\eta}$ for a certain A, where if A is defined, then $A \subseteq \mathcal{L}$. So it suffices to shown definedness. By the IH for the first premise, $[\![T]\!]_{\sigma,\rho} \in \mathcal{R}$. This means that if $E \in [\![T]\!]_{\sigma,\rho}$, $\zeta(E)$ is defined. We can then apply the IH to the second premise, since $\sigma[x \mapsto \zeta(E)] \in [\![\Gamma, x : T]\!]$, to obtain definedness of $[\![T']\!]_{\sigma[x\mapsto\zeta(E),\rho}$.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \star}{\Gamma \vdash \forall\, x{:}T.\, T' \Rightarrow \star}$$

By the IH for the second premise, $[\![T_2]\!]_{\sigma[x\mapsto\zeta(E)],\rho} \in \mathcal{R}$, for every $E \in [\![T_1]\!]_{\sigma,\rho}$ where $[\![T_1]\!]_{\sigma,\rho} \in \mathcal{R}$. By the IH for the first premise, we indeed have $[\![T_1]\!]_{\sigma,\rho} \in \mathcal{R}$. So if $[\![T_1]\!]_{\sigma,\rho}$ is non-empty, then the intersection of all the sets $[\![T_2]\!]_{\sigma[x\mapsto\zeta(E)],\rho}$ where $E \in [\![T_1]\!]_{\sigma,\rho}$ is a reducibility candidate, since each of those sets is. By the semantics of $\forall$-types quantifying over terms, this is sufficient. If $[\![T_1]\!]_{\sigma,\rho}$ is empty, then the interpretation of the $\forall$-type is $[\mathcal{L}]_{c\beta\eta}$ by the definition of $\cap_\star$, and this is in $\mathcal{R}$.

Case:

$$\frac{\Gamma \vdash \kappa \quad \Gamma, X : \kappa \vdash T \Rightarrow \star}{\Gamma \vdash \forall\, X{:}\kappa.\, T \Rightarrow \star}$$

Similarly to the previous case: by the IH for the second premise, $[\![T_2]\!]_{\sigma,\rho[X\mapsto S} \in \mathcal{R}$, for every $S \in [\![\kappa]\!]_{\sigma,\rho}$. By the IH part for the first premise, $[\![\kappa]\!]_{\sigma,\rho}$ is defined. So the intersection of all the sets $[\![T_2]\!]_{\sigma,\rho[X\mapsto S]}$ where $S \in [\![\kappa]\!]_{\sigma,\rho}$ is a reducibility candidate, since each of those sets is. The intersection is nonempty, since $[\![\kappa]\!]_{\sigma,\rho}$ is (as stated in a lemma above). By the semantics of $\forall$-types quantifying over types, this is sufficient.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \star}{\Gamma \vdash \iota\, x{:}T.\, T' \Rightarrow \star}$$

The set $[\![\iota x : T.T']\!]_{\sigma,\rho}$ is explicitly defined to be a subset of $[\![T]\!]_{\sigma,\rho}$, which is in $\mathcal{R}$, by the IH applied to the first premise. Since for any $A \subseteq \mathcal{L}$, $[A]_{c\beta\eta}$ is in $\mathcal{R}$, to show that $[\![\iota x : T.T']\!]_{\sigma,\rho}$ is also in $\mathcal{R}$ it suffices to show definedness of $[\![T']\!]_{\sigma[x\mapsto\zeta(E)],\rho}\}$ (which is used in the predicate picking out the particular subset of $[\![T]\!]_{\sigma,\rho}$), for $E \in [\![T]\!]_{\sigma,\rho}$. For such E, $\zeta(E)$ is defined (since $[\![T]\!]_{\sigma,\rho} \in \mathcal{R}$ and hence $E \in [\![T]\!]_{\sigma,\rho}$ is nonempty) and in E, so $\sigma[x \mapsto \zeta(E)] \in [\![\Gamma, x : T]\!]$. So by the IH for the second premise, $[\![T']\!]_{\sigma[x\mapsto\zeta(E),\rho]}$ is defined.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \kappa}{\Gamma \vdash \lambda\, x{:}T.\, T' \Rightarrow \Pi\, x{:}T.\, \kappa}$$

By the semantics, $\llbracket \lambda x : T.T' \rrbracket_{\sigma,\rho}$ is $(E \in \llbracket T \rrbracket_{\sigma,\rho} \mapsto \llbracket T' \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho})$. We must show that this (meta-level) function is in $\llbracket \Pi x : T.\kappa \rrbracket_{\sigma,\rho}$. By the semantics of kinds, the latter quantity, if defined, is $(E \in \llbracket T \rrbracket_{\sigma,\rho} \to_{c\beta\eta} \llbracket \kappa \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho})$. By the IH for the first premise, $\llbracket T \rrbracket_{\sigma,\rho} \in \mathcal{R}$. So we must just show that for any $E \in \llbracket T \rrbracket_{\sigma,\rho}$, $\llbracket T' \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho} \in \llbracket \kappa \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho}$. But this follows by the IH for the second premise.

Case:

$$\frac{\Gamma \vdash \kappa \quad \Gamma, X : \kappa \vdash T' \Rightarrow \kappa'}{\Gamma \vdash \lambda\, X \!:\! \kappa.\, T' \Rightarrow \Pi\, X \!:\! \kappa.\, \kappa'}$$

This case is an easier version of the previous one. It suffices to assume an arbitrary $S \in \llbracket \kappa \rrbracket_{\sigma,\rho}$ and show $\llbracket T' \rrbracket_{\sigma,\rho[X\mapsto S]} \in \llbracket \kappa' \rrbracket_{\sigma,\rho[X\mapsto S]}$. But this follows by the IH applied to the second premise. And we have definedness of $\llbracket \kappa \rrbracket_{\sigma,\rho}$ by the IH for the first premise.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \Pi\, x \!:\! T'.\, \kappa \quad \Gamma \vdash t \Leftarrow T'}{\Gamma \vdash T\ t \Rightarrow [t/x]^{T'}\ \kappa}$$

By the IH for the first premise, $\llbracket T \rrbracket_{\sigma,\rho} \in \llbracket \Pi x : T'.\kappa \rrbracket_{\sigma,\rho}$. By the semantics of Π-kinds, this means that $\llbracket T \rrbracket_{\sigma,\rho}$ is a function which given any $E \in \llbracket T' \rrbracket_{\sigma,\rho}$, will produce a result in $\llbracket \kappa \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho}$. By the semantics of type applications, $\llbracket T\ t \rrbracket_{\sigma,\rho}$ is equal to $\llbracket T \rrbracket_{\sigma,\rho}([\sigma|t|]_{c\beta\eta})$. This is defined, since $[\sigma|t|]_{c\beta\eta} \in \llbracket T' \rrbracket_{\sigma,\rho}$, by the IH for the second premise; note that $\llbracket T' \rrbracket_{\sigma,\rho}$ is defined since otherwise $\llbracket \Pi x : T'.\kappa \rrbracket_{\sigma,\rho}$ would not be defined. The result of applying the function is thus indeed in $\llbracket [t/x]^{T'}\ \kappa \rrbracket_{\sigma,\rho}$, since $|\chi\ T'\ \text{-}\ t| = |t|$ (recall the shorthand $[t/x]^{T'} = [\chi\ T'\ \text{-}\ t/x]$), and with Lemma 12 we have that the interpretation equals $\llbracket \kappa \rrbracket_{\sigma[x\mapsto\zeta([\sigma|t|]_{c\beta\eta})],\rho}$ (the codomain of the function being applied).

Case:

$$\frac{\Gamma \vdash T_1 \Rightarrow \Pi\, X \!:\! \kappa_2.\, \kappa_1 \quad \Gamma \vdash T_2 \Rightarrow \kappa_2' \quad \kappa_2 \cong \kappa_2'}{\Gamma \vdash T_1 \cdot T_2 \Rightarrow [T_2/X]\kappa_1}$$

By the IH applied to the first premise, $\llbracket T_1 \rrbracket_{\sigma,\rho} \in \llbracket \Pi\, X : \kappa_2.\, \kappa_1 \rrbracket_{\sigma,\rho}$. By the semantics of Π-kinds, this means that for any $S \in \llbracket \kappa_2 \rrbracket_{\sigma,\rho}$, $\llbracket T_1 \rrbracket_{\sigma,\rho}\ S$ is in $\llbracket \kappa_1 \rrbracket_{\sigma,\rho[X\mapsto S]}$. By the IH for the second premise, we have $\llbracket T_2 \rrbracket \in \llbracket \kappa_2' \rrbracket_{\sigma,\rho}$, and by the IH for the third premise, we have $\llbracket \kappa_2 \rrbracket_{\sigma,\rho} = \llbracket \kappa_2' \rrbracket_{\sigma,\rho}$. So we get $\llbracket T_1 \rrbracket_{\sigma,\rho}(\llbracket T_2 \rrbracket_{\sigma,\rho}) \in \llbracket \kappa_1 \rrbracket_{\sigma,\rho[X\mapsto\llbracket T' \rrbracket_{\sigma,\rho}]}$, which suffices by Lemma 13.

Case:

$$\frac{FV(t\ t') \subseteq dom(\Gamma)}{\Gamma \vdash \{t \simeq t'\} : \star}$$

Either $\sigma|t| =_{c\beta\eta} \sigma|t'|$ or not. Either way, the interpretation is defined and in $\mathcal{R}$, since $FV(t\ t') \subseteq dom(\sigma)$ (as an easy consequence of $(\sigma, \rho) \in \llbracket \Gamma \rrbracket$).

A.3 Proof of parts (3) and (4)

Case:

$$\frac{(x : T) \in \Gamma}{\Gamma \vdash x \Rightarrow T}$$

This follows from the definition of $\llbracket \Gamma \rrbracket$.

Case:

$$\frac{\Gamma \vdash t \Rightarrow T' \quad T' \cong T}{\Gamma \vdash t \Leftarrow T}$$

By the IH applied to the first premise, we have $[\sigma|t|]_{c\beta\eta} \in \llbracket T' \rrbracket_{\sigma,\rho} \in \mathcal{R}$. By assumption, $\llbracket T \rrbracket_{\sigma,\rho} \in \mathcal{R}$, and so by the IH applied to the second premise, we have $[\sigma|t|]_{c\beta\eta} \in \llbracket T' \rrbracket_{\sigma,\rho} = \llbracket T \rrbracket_{\sigma,\rho}$.

Case:

$$\frac{T \rightsquigarrow_\beta^* \Pi\, x\!:\!T_1.\, T_2 \quad \Gamma, x : T_1 \vdash t \Leftarrow T_2}{\Gamma \vdash \lambda\, x.\, t \Leftarrow T}$$

To show $[\sigma\lambda x.|t|]_{c\beta\eta} \in \llbracket \Pi\, x\!:\!T_1.\, T_2 \rrbracket_{\sigma,\rho}$ (noting that the latter is defined and in $\mathcal{R}$ by assumption), it suffices to assume an arbitrary $E \in \llbracket T_1 \rrbracket_{\sigma,\rho}$, and show $[[\zeta(E)/x]\sigma|t|]_{c\beta\eta} \in \llbracket T_2 \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho}$. By the IH, we have $[\sigma[x \mapsto \zeta(E)]|t|]_{c\beta\eta} \in \llbracket T_2 \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho}$. But $[\sigma[x \mapsto \zeta(E)]t]_{c\beta\eta} = [[\zeta(E)/x]\sigma|t|]_{c\beta\eta}$, so this is sufficient.

Case:

$$\frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \Pi\, x\!:\!T'.\, T \quad \Gamma \vdash t' \Leftarrow T'}{\Gamma \vdash t\ t' \Rightarrow [t'/x]^{T'} T}$$

By the IH applied to the first premise, $[\sigma|t|]_{c\beta\eta} \in \llbracket \Pi x : T'.T \rrbracket_{\sigma,\rho} \in \mathcal{R}$. This means that there exists a λ-abstraction $\lambda x.\hat{t}$ such that $\lambda x.\hat{t} =_{c\beta\eta} \sigma|t|$, by the semantics of Π-types. Furthermore, for any $E \in \llbracket T' \rrbracket_{\sigma,\rho}$, $[[\zeta(E)/x]\hat{t}]_{c\beta\eta} \in \llbracket T \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho}$. By the IH applied to the second premise, $[\sigma|t'|]_{c\beta\eta} \in \llbracket T' \rrbracket_{\sigma,\rho}$, so we can instantiate the quantifier in the previous formula to obtain

$$[[\zeta([\sigma|t'|]_{c\beta\eta})/x]\hat{t}]_{c\beta\eta} \in \llbracket T \rrbracket_{\sigma[x\mapsto\zeta([\sigma|t'|]_{c\beta\eta})],\rho}$$

By Lemma 12, this is equivalent to

$$[[\zeta([\sigma|t'|]_{c\beta\eta})/x]\hat{t}]_{c\beta\eta} \in \llbracket [t'/x]T_2 \rrbracket_{\sigma,\rho}$$

Since $\sigma|t\ t'| =_{c\beta\eta} (\lambda x.\hat{t})\ \sigma|t'| =_{c\beta\eta} [[\zeta([\sigma|t'|]_{c\beta\eta})/x]\hat{t}$, this is sufficient.

Case:

$$\frac{T' \rightsquigarrow_\beta^* \forall\, X\!:\!\kappa.\, T \quad \Gamma, X : \kappa \vdash t \Leftarrow T}{\Gamma \vdash \Lambda\, X.\, t \Leftarrow T'}$$

By the IH, $[\sigma|t|]_{c\beta\eta} \in \llbracket T \rrbracket_{\sigma,\rho[X\mapsto S]}$, for all $S \in \llbracket \kappa \rrbracket_{\sigma,\rho}$. This is sufficient to prove $[\sigma|\Lambda\, X.\, t|]_{c\beta\eta} \in \llbracket \forall X : \kappa.T \rrbracket_{\sigma,\rho}$, by the semantics of $\forall$-types and definition of erasure.

Case:

$$\frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \forall\, X\!:\!\kappa.\, T \quad \Gamma \vdash T' \Rightarrow \kappa' \quad \kappa' \cong \kappa}{\Gamma \vdash t \cdot T' \Rightarrow [T'/X]T}$$

By the semantics of $\forall$-types and the IH applied to the first premise, we have $[\sigma|t|]_{c\beta\eta} \in \llbracket T \rrbracket_{\sigma,\rho[X\mapsto S]}$, for all $S \in \llbracket \kappa \rrbracket_{\sigma,\rho}$. By applying the IH twice, once to the second premise and once to the third, we have, we have $\llbracket T' \rrbracket_{\sigma,\rho} \in \llbracket \kappa \rrbracket_{\sigma,\rho}$. So, can derive $[\sigma|t|]_{c\beta\eta} \in \llbracket T \rrbracket_{\sigma,\rho[X\mapsto\llbracket T' \rrbracket_{\sigma,\rho}]}$. By Lemma 13, this is equivalent to the required $[\sigma|t|]_{c\beta\eta} \in \llbracket [T'/X]T \rrbracket_{\sigma,\rho}$, using also the definition of erasure.

Case:

$$\frac{T \rightsquigarrow_\beta^* \forall\, x\!:\!T_1.\, T_2 \quad \Gamma, x : T_1 \vdash t \Leftarrow T_2 \quad x \notin FV(|t|)}{\Gamma \vdash \Lambda\, x.\, t \Leftarrow T}$$

By the IH applied to the second premise, we have $[\sigma[x \mapsto \zeta(E)]|t|]_{c\beta\eta} \in \llbracket T_2 \rrbracket_{\sigma[x\mapsto\zeta(E)],\rho}$, for any $E \in \llbracket T_1 \rrbracket_{\sigma,\rho}$. This is because $\llbracket T_1 \rrbracket_{\sigma,\rho} \in \mathcal{R}$, since $\llbracket \forall\, x\!:\!T_1.\, T_2 \rrbracket_{\sigma,\rho}$ is in $\mathcal{R}$ and hence defined, by assumption. Since $x \notin FV(|t|)$, we know $[[\sigma[x \mapsto \zeta(E)]|t|]_{c\beta\eta} = [\sigma|t|]_{c\beta\eta}$. By the semantics of $\forall$-types and definition of erasure, this suffices to show the desired conclusion.

Case:

$$\frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \forall\, x\!:\!T'.\, T \quad \Gamma \vdash t' \Leftarrow T'}{\Gamma \vdash t\ \text{-}t' \Rightarrow [t'/x]^{T'} T}$$

The result follows easily by the IH applied to the premises, the semantics of $\forall$-types, definition of erasure, and Lemma 12.

Case:

$$\frac{\begin{array}{cc} T \rightsquigarrow_\beta^* \iota\, x\!:\!T_1.\, T_1 & \Gamma \vdash t_1 \Leftarrow T_1 \\ \Gamma \vdash t_2 \Leftarrow [t/x]^{T_1}\; T_2 & |t_1| =_{\beta\eta} |t_2| \end{array}}{\Gamma \vdash [t_1, t_2] \Leftarrow T}$$

By the IH, we have $[\sigma|t_1|]_{c\beta\eta} \in [\![T_1]\!]_{\sigma,\rho}$ and $[\sigma|t_2|]_{c\beta\eta} \in [\![[t_1/x]T_2]\!]_{\sigma,\rho}$. By Lemma 12, the latter is equivalent to $[\sigma|t|]_{c\beta\eta} \in [\![T_2]\!]_{\sigma[x\mapsto\zeta([\sigma|t|]_{c\beta\eta}),\rho}$. These two facts about $[\sigma|t|]_{c\beta\eta}$ are sufficient, by the semantics of ι-types, for the desired conclusion, using also the fact (from the fourth premise) that $\sigma|t| =_{c\beta\eta} \sigma|t'|$.

Case:

$$\frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \iota\, x\!:\!T.\, T'}{\Gamma \vdash t.1 \Rightarrow T}$$

The desired conclusion follows easily from the IH and the semantics of ι-types.

Case:

$$\frac{\Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \iota\, x\!:\!T.\, T'}{\Gamma \vdash t.2 \Rightarrow [t.1/x]T'}$$

Similar to the previous case, additionally using Lemma 12.

Case:

$$\frac{T \rightsquigarrow_\beta^* \{t_1 \simeq t_2\} \quad FV(t') \subseteq dom(\Gamma) \quad |t_1| =_{\beta\eta} |t_2|}{\Gamma \vdash \beta\{t'\} \Leftarrow T}$$

By the third premise $|t_1| =_{\beta\eta} |t_2|$ it follows that $\sigma|t_1| =_{\beta\eta} \sigma|t_2|$, and also by the second premise and the fact that $\sigma \in [\![\Gamma]\!]$ it follows that $FV(t_1\ t_2) \subseteq dom(\sigma)$. Also, we see $\sigma|t'|$ is a closed term. So, $[\sigma|t'|]_{c\beta\eta} \in [\![\{t_1 \simeq t_2\}]\!]_{\sigma,\rho}$ follows directly from the semantics of equality types.

Case:

$$\frac{\Gamma \vdash t \Rightarrow \{\lambda\, x.\, \lambda\, y.\, x \simeq \lambda\, x.\, \lambda\, y.\, y\}}{\Gamma \vdash \delta\ \text{-}\ t \Leftarrow T}$$

By the semantics of equality types, $[\sigma|t|]_{c\beta\eta}$ cannot be in the interpretation of the equation in the premise, since the two terms in question are closed and not $\beta\eta$-equal. By the IH applied to the first premise, however, $[\sigma|t|]_{c\beta\eta}$ is in the interpretation of that equation. This is a contradiction.

Case:

$$\frac{\begin{array}{ccc} \Gamma \vdash t' \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \{t_1 \simeq t_2'\} & FV(t_2) \subseteq dom(\Gamma) & |t_2'| =_{\beta\eta} |t_2| \\ \Gamma \vdash [t_2/x]\ T \Rightarrow \star & \Gamma \vdash t \Rightarrow T' & T' \cong [t_1/x]T \end{array}}{\Gamma \vdash \rho\ t'\ @x\langle t_2\rangle.T\ \text{-}\ t \Rightarrow [t_2/x]\ T}$$

By the IH applied to the first premise in the first row, $\sigma|t_1| =_{c\beta\eta} \sigma|t_2'|$. With the second and third premise, and from the assumption $\sigma \in [\![\Gamma]\!]$, we have $\sigma|t_2'| =_{c\beta\eta} \sigma|t_2|$ (and $\sigma|t_2|$ is closed), so we obtain $[\sigma|t_1|]_{c\beta\eta} = [\sigma|t_2|]_{c\beta\eta}$. By the IH applied to the first premise of the second row, $[\![[t_2/x]T]\!]_{\sigma,\rho} \in \mathcal{R}$ and so is defined. By the IH applied to the first premise in the second row, $[\sigma|t|]_{c\beta\eta} \in [\![T']\!]_{\sigma,\rho}$. By the IH applied to the second premise in the second row, $[\![T']\!]_{\sigma,\rho} = [\![[t_1/x]T]\!]_{\sigma,\rho}$. The result then follows by applying Lemma 12.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma \vdash t \Leftarrow T}{\Gamma \vdash \chi\ T\ \text{-}\ t\ \Rightarrow T}$$

We apply the IH for the first premise to get that $[\![T]\!]_{\sigma,\rho}$ is in $\mathcal{R}$ and hence defined. Using this, we apply the IH on the second premise to get $[\sigma|t|]_{c\beta\eta} \in [\![T]\!]_{\sigma,\rho}$, which by the definition of erasure is what we must show.

Case:

$$\frac{\begin{array}{c} \Gamma \vdash t \overset{\rightsquigarrow_\beta^*}{\Rightarrow} \{t_1 \simeq t_2\} \qquad\qquad \Gamma \vdash t' \Leftrightarrow T \\ |t_1| =_{\beta\eta} |t'| \qquad |t_2| =_{\beta\eta} |t''| \quad FV(t'') \subseteq dom(\Gamma) \end{array}}{\Gamma \vdash \varphi\ t\ \text{-}\ t'\ \{t''\} \Leftrightarrow T}$$

By the IH for the first premise, $\sigma|t_1| =_{c\beta\eta} \sigma|t_2|$. Using the first and second premise of the second row, we obtain $\sigma|t'| =_{c\beta\eta} \sigma|t''|$. By the IH for the second premise, $[\sigma|t'|]_{c\beta\eta} \in [\![T]\!]_{\sigma,\rho}$ (in the case of type synthesis, the IH also tells us $[\![T]\!]_{\sigma,\rho}$ is defined). This suffices for the desired conclusion, using also the definition of erasure ($|\varphi\ t\ \text{-}\ t'\{t''\}| = |t''|$).

Proof of part (5)

Case:

$$\frac{T \rightsquigarrow_\beta^* T_1 \quad T' \rightsquigarrow_\beta^* T_2 \quad T_1 \cong^t T_2}{T \cong T'}$$

By Lemma 14, we have

$$\begin{array}{rcl} [\![T]\!]_{\sigma,\rho} & = & [\![T_1]\!]_{\sigma,\rho} \\ [\![T']\!]_{\sigma,\rho} & = & [\![T_2]\!]_{\sigma,\rho} \end{array}$$

By the IH for the third premise, we have $[\![T_1]\!]_{\sigma,\rho} = [\![T_2]\!]_{\sigma,\rho}$, which suffices.

Case:

$$\frac{T \cong^t T'}{T \cong T'}$$

By the IH.

Case:

$$\frac{T \cong^t T' \quad |t| =_{\beta\eta} |t'|}{T\ t \cong^t T'\ t'}$$

By the semantics, $[\![T\ t]\!]_{\sigma,\rho} = [\![T]\!]_{\sigma,\rho}([\sigma|t|]_{c\beta\eta})$. By the second premise and the IH for the first premise, this equals $[\![T']\!]_{\sigma,\rho}([\sigma|t'|]_{c\beta\eta})$, as required.

Case:

$$\frac{|t_1| =_{\beta\eta} |t_1'| \quad |t_2| =_{\beta\eta} |t_2'|}{\{t_1 \simeq t_2\} \cong^t \{t_1' \ \simeq t_2'\}}$$

This follows easily from the premises and the semantics of equality types.

Proof of Part (6)

The convertibility relation for kinds consists entirely of congruential rules for quantification over types and kinds.

□

B Proof of Theorem 4

Proof (of Theorem 4). Theorem 1 implies that since $t'\ t$ is closed and of type $\Pi x : T_1.\ T_2$, we have $[|t'\ t|]_{c\beta\eta} \in [\![\Pi x : T_1.\ T_2]\!]_{\cdot,\rho}$, where $[|t'\ t|]_{c\beta\eta}$ is the set of closed terms which are $\beta\eta$-equivalent to $|t'\ t|$; and $(\cdot, \rho) \in [\![\Gamma]\!]$ gives interpretations $\cdot$ for term- and ρ for type-variables in Γ. By the semantics of types defined in Figure 7, the interpretation of a Π-type consists of sets of the form $[\lambda x.\,t']_{c\beta\eta}$. So we have that $|t'\ t|$ is $\beta\eta$-equivalent to $\lambda x.\,t'$ for some x, t'. Since $|t'\ t|$ is β-equivalent to t, we know $t =_{\beta\eta} \lambda x.\,t'$. It is then an easy consequence of the standardization theorem for untyped lambda calculus that $|t|$ is call-by-name normalizing (cf. [7]). □

C Proof of Theorems 6 and 7

First a few lemmas (easy proofs omitted):

Lemma 15. *Let κ_1, κ_1' be kinds such that $\kappa_1 \cong \kappa_1'$.*

- *If $\Gamma_1, X : \kappa_1, \Gamma_2 \vdash \kappa_2$ and $\Gamma_1 \vdash \kappa_1'$ then $\Gamma_1, X : \kappa_1', \Gamma_2 \vdash \kappa_2$*
- *If $\Gamma_1, X : \kappa_1, \Gamma_2 \vdash T \Rightarrow \kappa_2$ and $\Gamma_1 \vdash \kappa_1'$ then $\Gamma_1, X : \kappa_1', \Gamma_2 \vdash T \Rightarrow \kappa_2$ for some $\kappa_2' \cong \kappa_2$*
- *If $\Gamma_1, X : \kappa_1, \Gamma_2 \vdash t \Leftrightarrow T$ and $\Gamma_1 \vdash \kappa_1'$ then $\Gamma_1, X : \kappa_1', \Gamma_2 \vdash t \Leftrightarrow T$.*

Proof. By induction on the assumed derivation. We show a few interesting cases.

Case:

$$\frac{(X : \kappa) \in \Gamma_1, X_1 : \kappa_1, \Gamma_2}{\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash X \Rightarrow \kappa}$$

We have two subcases to consider. If $X = X_1$, then by the rule we obtain $\Gamma_1, X_1 : \kappa_1', \Gamma_2 \vdash X_1 \Rightarrow \kappa_1'$, as desired. Otherwise, by the rule we obtain $\Gamma_1, X_1 : \kappa_1', \Gamma_2 \vdash X \Rightarrow \kappa$.

Case:

$$\frac{\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash \kappa \quad \Gamma_1, X_1 : \kappa_1, \Gamma_2, X : \kappa \vdash T' \Rightarrow \kappa'}{\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash \lambda\, X{:}\kappa.\, T' \Rightarrow \Pi\, X{:}\kappa.\, \kappa'}$$

By the IH on the first premise, $\Gamma_1, X_1 : \kappa_1', \Gamma_2 \vdash \kappa$. By the IH on the second premise, $\Gamma_1, X_1 : \kappa_1', \Gamma_2, X : \kappa \vdash T' \Rightarrow \kappa''$ for some $\kappa'' \cong \kappa'$. By the rule, $\Gamma_1, X_1\kappa_1', \Gamma_2 \vdash \lambda\, X{:}\kappa.\, T' \Rightarrow \Pi\, X{:}\kappa.\, \kappa''$, which is convertible with the original kind synthesized for this type.

Case:

$$\frac{\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash T \Rightarrow \Pi\, X{:}\kappa'.\, \kappa \quad \Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash T' \Rightarrow \kappa'' \quad \kappa' \cong \kappa''}{\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash T \cdot T' \Rightarrow [T'/X]\kappa}$$

By the IH on the first premise, $\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash T \Rightarrow \Pi\, X{:}\kappa_2'.\, \kappa_2$ for some $\kappa_2' \cong \kappa'$, $\kappa_2 \cong \kappa$. By the IH on the second premise, $\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash T' \Rightarrow \kappa_2''$ for some $\kappa_2'' \cong \kappa''$. By transitivity, $\kappa_2' \cong \kappa_2''$. By the rule, $\Gamma_1, X_1 : \kappa_1', \Gamma_2 \vdash T \cdot T' \Rightarrow [T'/X]\kappa_2$, with this kind congruent to $[T'/X]\kappa$.

Case:

$$\frac{\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash t \Rightarrow \forall\, X{:}\kappa.\, T \quad \Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash T' \Rightarrow \kappa' \quad \kappa' \cong \kappa}{\Gamma_1, X_1 : \kappa_1, \Gamma_2 \vdash t \cdot T' \Rightarrow [T'/X]T}$$

By the IH on the first premise, $\Gamma_1, X_1 : \kappa_1', \Gamma_2 \vdash t \Rightarrow \forall\, X : \kappa.\, T$. By the IH on the second premise, $\Gamma_1, X_1 : \kappa_1', \Gamma_2 \vdash T' \Rightarrow \kappa''$ for some $\kappa'' \cong \kappa'$. With the third premise and transitivity of congruence, we have $\kappa'' \cong \kappa$. We apply the rule and conclude. □

Corollary 16. *If $\vdash \Gamma_1, X : \kappa_1, \Gamma_1$ and $\Gamma_1 \vdash \kappa_1'$ with $\kappa_1' \cong \kappa$ then $\vdash \Gamma_1, X : \kappa_1', \Gamma_2$.*

Lemma 17. *Below, each statement seperately universally quantifies over meta-variables, and it is assumed that typing contexts occuring in assumed derivations are well-formed.*

1. ***Kinds:***
   - *If* $\Gamma_1, x : T, \Gamma_2 \vdash \kappa$ *and* $\Gamma_1 \vdash t \Leftarrow T$ *then* $\Gamma_1, [t/x]\Gamma_2 \vdash [t/x]\kappa$
   - *If* $\Gamma_1, X : \kappa', \Gamma_2 \vdash \kappa$ *and* $\Gamma_1 \vdash T \Rightarrow \kappa'$ *then* $\Gamma_1, [T/X]\Gamma_2 \vdash [T/X]\kappa$
2. ***Types***
   - *If* $\Gamma_1, x : T', \Gamma_2 \vdash T \Rightarrow \kappa$ *and* $\Gamma_1 \vdash t \Rightarrow T'$ *then* $\Gamma_1, [t/x]\Gamma_2 \vdash [t/x]T \Rightarrow [t/x]\kappa$
   - *If* $\Gamma_1, X : \kappa_2, \Gamma_2 \vdash T_1 \Rightarrow \kappa_1$ *and* $\Gamma_1 \vdash T_2 \Rightarrow \kappa_2$ *then* $\Gamma_1, [T_2/X]\Gamma_2 \vdash [T_2/X]T_1 \Rightarrow [T_2/X]\kappa_1$
3. ***Terms:***
   - *If* $\Gamma_1, x : T', \Gamma_2 \vdash t \Leftrightarrow T$ *and* $\Gamma_1 \vdash t' \Rightarrow T'$ *then* $\Gamma_1, [t'/x]\Gamma_2 \vdash [t'/x]t \Rightarrow [t'/x]T$
   - *If* $\Gamma_1, X : \kappa, \Gamma_2 \vdash t \Leftrightarrow T'$ *and* $\Gamma_1 \vdash T \Rightarrow \kappa$ *then* $\Gamma_1, [T/X]\Gamma_2 \vdash [T/X]t \Rightarrow [T/X]T'$

Proof. By mutual induction on the assumed derivations. We only show a few interesting cases, and we omit type annotations on substitutions when these are clear from context.

Case:

$$\frac{(X_1 : \kappa_1) \in \Gamma_1, X : \kappa, \Gamma_2}{\Gamma_1, X : \kappa, \Gamma_2 \vdash X_1 \Rightarrow \kappa_1}$$

We have two cases. If $X_1 = X$, then $\kappa_1 = \kappa$ and by assumption $\Gamma_1 \vdash T \Rightarrow \kappa$, and the desired result holds by weakening. Otherwise, either $(X_1 : \kappa_1) \in \Gamma_1$ and $X \notin FV(\kappa_1)$, or else $(X_1 : \kappa_1) \in \Gamma_2$. Either way, we have $\Gamma_1, [T/X]\Gamma_2 \vdash X_1 \Rightarrow [T/X]\kappa_1$.

Case:

$$\frac{\Gamma_1, x : T, \Gamma_2 \vdash T_2 \Rightarrow \Pi\, x_1 {:} T_1.\, \kappa \quad \Gamma_1, x : T, \Gamma_2 \vdash t_1 \Leftarrow T_1}{\Gamma_1, x : T, \Gamma_2 \vdash T_2\ t_1 \Rightarrow [t_1/x_1]^{T_1}\kappa}$$

By the IH, $\Gamma_1, [t/x]\Gamma_2 \vdash [t/x]T_2 \Rightarrow \Pi\, x_1 : [t/x]T_1.\, [t/x]\kappa$ and $\Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t_1 \Leftarrow [t/x]T_1$. By the rule, we have $\Gamma_1, [t/x]\Gamma_2 \vdash [t/x]T_2\ [t/x]t_1 \Rightarrow [[t/x]t_1/x_1][t/x]\kappa$, where the synthesized kind is equal to the desired $[t/x][t_1/x_1]\kappa$.

Case:

$$\frac{FV(t_1\ t_2) \subseteq dom(\Gamma_1, x : T, \Gamma_2)}{\Gamma_1, x : T, \Gamma_2 \vdash \{t_1 \simeq t_2\} : \star}$$

It suffices to show that $FV([t/x]t_1\ [t/x]t_2) \subseteq dom(\Gamma_1, [t/x]\Gamma_2)$. It is clear x is not a free variable of this expression, that the free variables of t are declared in Γ_1, and by assumption the other free variables of it are declared in $\Gamma_1, [t/x]\Gamma_2$.

Case:

$$\frac{(x_1 : T_1) \in \Gamma_1, x : T, \Gamma_2}{\Gamma_1, x : T, \Gamma_2 \vdash x_1 \Rightarrow T_1}$$

We elaborate on the case where $x_1 = x$. It is important that we assumed that $\Gamma_1 \vdash t \Rightarrow T$ (as opposed to having its type checked), as we may now replace the given rule with this assumed derivation without changing the definition of substitution.

Case:

$$\frac{FV(t') \subseteq \mathit{dom}(\Gamma_1, x : T, \Gamma_2) \quad |t_1| =_{\beta\eta} |t_2|}{\Gamma_1, x : T, \Gamma_2 \vdash \beta\{t'\} \Leftarrow \{t_1 \simeq t_2\}}$$

From our assumptions we may conclude $FV([t/x]t) \subseteq \mathit{dom}(\Gamma_1, [t/x]\Gamma_2)$, and from the second premise that $|[t/x]t_1| =_{\beta\eta} |[t/x]t_2|$. Thus, $\Gamma_1, [t/x]\Gamma_2 \ \vdash \beta\{[t/x]t'\} \Leftarrow \{[t/x]t_1 \simeq [t/x]t_2\}$.

Case:

$$\frac{\begin{array}{cc} \Gamma_1, x : T, \Gamma_2 \vdash t'' \Rightarrow \{t_1 \simeq t_2\} & \Gamma_1, x : T, \Gamma_2 \vdash [\beta\{t_2\}/x_1]^{\mathtt{Top}}\ T_1 \Rightarrow \star \\ \Gamma_1, x : T, \Gamma_2 \vdash t' \Rightarrow T' & T' \cong [t_1/x_1]T_1 \end{array}}{\Gamma_1, x : T, \Gamma_2 \vdash \rho\ t''\ @x_1.T_1\ \text{-}\ t' \Rightarrow [\beta\{t_2\}/x_1]^{\mathtt{Top}}\ T_1}$$

From the IH, we have that $\Gamma_1, [t/x]\Gamma_2 \vdash [t/x]t'' \Rightarrow \{[t/x]t_1 \simeq [t/x]t_2\}$, that $\Gamma_1, [t/x]\Gamma_2 \vdash [t/x][\beta\{t_2\}/x]T_1 \Rightarrow \star$, and that $\Gamma_1, [t/x]\Gamma_2 \ \vdash \ [t/x]t' \ \Rightarrow \ [t/x]T'$. From the last premise, we may conclude that $[t/x]T' \cong [t/x][t_1/x_1]T_1$. Applying the rule and permuting substitutions gives us the desired result (note for example that $[t/x][t_1/x_1]T_1 = [[t/x]t_1/x_1]T_1$). □

Corollary 18.

- *If* $\vdash \Gamma_1, x : T, \Gamma_2$ *and* $\Gamma_1 \vdash t \Rightarrow T$ *then* $\vdash \Gamma_1, [t/x]\Gamma_2$
- *If* $\vdash \Gamma_1, X : \kappa, \Gamma_2$ *and* $\Gamma_1 \vdash T \Rightarrow \kappa$ *then* $\vdash \Gamma_1, [T/X]\Gamma_2$.

Proof. By induction on the assumed derivation, appealing to Lemma 17 at each step. □

Lemma 19. *If* $\Gamma \vdash t \Leftarrow T_1$ *where* $\Gamma \vdash T_1 \Rightarrow \kappa_1$*, and we have* $\Gamma \vdash T_2 \Rightarrow \kappa_2$ *with* $T_1 \cong T_2$ *and* $\kappa_1 \cong \kappa_2$*, then* $\Gamma \vdash t \Leftarrow T_2$.

Proof (of Theorem 6). We may rule out the cases where T is a variable or formed by a type constructor, as this would contradict the assumption that $T \rightsquigarrow_\beta T'$ for some T' (recall that $\rightsquigarrow_\beta$ is call-by-name reduction for types). We omit type annotations from substitutions when they are clear from the context.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \Pi\, x\!:\!T_1.\, \kappa_1 \quad \Gamma \vdash t \Leftarrow T_1}{\Gamma \vdash T\ t \Rightarrow [t/x]^{T_1}\kappa_1}$$

There are two subcases to consider for the derivation of $T\ t \rightsquigarrow_\beta T'$. In the first case, we have $T \rightsquigarrow_\beta T''$ for some T'' (so $T' = T''\ t$). By the IH, we have $\Gamma \vdash T'' \Rightarrow \Pi\, x\!:\!T_1'.\, \kappa_1'$ for some $T_1' \cong T_1$ and $\kappa_1' \cong \kappa_1$ (we have by assumption the kind of T'' must be convertible with $\Pi\, x\!:\!T_1.\, \kappa_1$). By Lemma 19, $\Gamma \vdash t \Leftarrow T_1'$. By the rule, $\Gamma \vdash T''\ t \Rightarrow [t/x]\kappa_1'$, and the synthesized kind is clearly convertible with $[t/x]\kappa_1$.

In the other subcase, the subject of kinding is of the form $(\lambda\, x\!:\!T_1.\, T'')\ t$ and our assumed reduction is to $[t/x]T''$. By inversion of the kinding derivation, we have $\Gamma, x : T_1 \vdash T'' \Rightarrow \kappa_1$. By Lemma 17, we have $\Gamma \vdash [t/x]T'' \Rightarrow [t/x]\kappa_1$.

Case:

$$\frac{\Gamma \vdash T_1 \Rightarrow \Pi\, X\!:\!\kappa_2.\, \kappa_1 \quad \Gamma \vdash T_2 \Rightarrow \kappa_2' \quad \kappa_2 \cong \kappa_2'}{\Gamma \vdash T_1 \cdot T_2 \Rightarrow [T_2/X]\kappa_1}$$

There are two subcases to consider for the derivation of $T_1 \cdot T_2 \rightsquigarrow_\beta T'$. In the first subcase, we have $T_1 \rightsquigarrow_\beta T_1''$ for some T_1'' (so $T' = T_1'' \cdot T_2$). By the IH, we have $\Gamma \vdash T_1'' \Rightarrow \Pi\, X\!:\!\kappa_2''.\, \kappa_1''$ for some $\kappa_2'' \cong \kappa_2$ and $\kappa_1'' \cong \kappa_1$ (we have by assumption the kind of T_1'' must be convertible with $\Pi\, X\!:\!\kappa_2.\, \kappa_1$). By transitivity we can conclude

$\kappa_2'' \cong \kappa_2$. By the rule, we have $\Gamma \vdash T_1'' \cdot T_2 \Rightarrow [T_2/X]\kappa_1''$, and the synthesized kind is clearly convertible with $[T_2/X]\kappa_1$.

In the other subcase, the subject of kinding is of the form $(\lambda\, X : \kappa_2.\, T_1'') \cdot T_2$ and our assumed reduction is to $[T_2/X]T_1''$. By inversion of the kinding derivation, we have $\Gamma, X : \kappa_2 \vdash T_1'' \Rightarrow \kappa_1$. By Lemma 17, we have $\Gamma \vdash [T_2/X]T_1'' \Rightarrow [T_2/X]\kappa_1$. □

Proof (of Theorem 7). By induction on the assumed derivation. We show a few interesting cases.

Case:

$$\frac{(X : \kappa) \in \Gamma}{\Gamma \vdash X \Rightarrow \kappa}$$

By an easy inductive argument on the assumption $\vdash \Gamma$ and weakening, we obtain $\Gamma \vdash \kappa$.

Case:

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma, x : T \vdash T' \Rightarrow \kappa}{\Gamma \vdash \lambda\, x : T.\, T' \Rightarrow \Pi\, x : T.\, \kappa}$$

By assumption, the first premise, and the context formation rules, we have $\vdash \Gamma, x : T$. By the IH, we have $\Gamma, x : T \vdash \kappa$. Thus we obtain $\Gamma \vdash \Pi\, x : T.\, \kappa$.

Case:

$$\frac{\Gamma \vdash T_1 \Rightarrow \Pi\, X : \kappa_2.\, \kappa_1 \quad \Gamma \vdash T_2 \Rightarrow \kappa_2' \quad \kappa_2 \cong \kappa_2'}{\Gamma \vdash T_1 \cdot T_2 \Rightarrow [T_2/X]\kappa_1}$$

By the IH on the first premise, we have $\Gamma \vdash \Pi\, X : \kappa_2.\, \kappa_1$, and by inversion this gives us $\Gamma \vdash \kappa_2$, which yields $\vdash \Gamma, X : \kappa_2$, and $\Gamma, X : \kappa_2 \vdash \kappa_1$. By the IH on the second premise, $\Gamma \vdash \kappa_2'$. By Lemma 15 using the third premise, $\Gamma, X : \kappa_2' \vdash \kappa_1$. By Lemma 17, $\Gamma \vdash [T_2/X]\kappa_1$.

Case:

$$\frac{\Gamma \vdash t \Rightarrow T \quad T' \rightsquigarrow_\beta^* T'}{\Gamma \vdash t \Rightarrow T'}$$

By the IH, $\Gamma \vdash T \Rightarrow \star$. By generalizing Theorem 6 to $\rightsquigarrow_\beta^*$, we have $\Gamma \vdash T' \Rightarrow \kappa'$ for some $\kappa' \cong \star$. This means that in fact $\kappa' = \star$, as desired.

Case:

$$\frac{\Gamma \vdash t \Rightarrow \Pi\, x : T'.\, T \quad \Gamma \vdash t' \Leftarrow T'}{\Gamma \vdash t\ t' \Rightarrow [t'/x]^{T'} T}$$

By the IH on the first premise, $\Gamma \vdash \Pi\, x : T'.\, T \Rightarrow \star$. By inversion of this, we obtain both $\Gamma \vdash T' \Rightarrow \star$, from which it follows by the χ rule and second premise that $\Gamma \vdash \chi\ T'\ \text{-}\ t \Rightarrow T'$, and that $\Gamma, x : T' \vdash T \Rightarrow \star$, from which we obtain $\Gamma \vdash [t/x]^{T'}\ T \Rightarrow \star$ using Lemma 17

Case:

$$\frac{\begin{array}{cc} \Gamma \vdash t' \Rightarrow \{t_1 \simeq t_2\} & \Gamma \vdash [\beta\{t_2\}/x]^{\texttt{Top}}\ T \Rightarrow \star \\ \Gamma \vdash t \Rightarrow T' & T' \cong [t_1/x]T \end{array}}{\Gamma \vdash \rho\ t'\ @x.T\ \text{-}\ t \Rightarrow [\beta\{t_2\}/x]^{\texttt{Top}}\ T}$$

The desired result is given by the second assumption (reading first left to right, then top to bottom).

Case:

$$\frac{\Gamma \vdash T \Rightarrow \star \quad \Gamma \vdash t \Leftarrow T}{\Gamma \vdash \chi\ T\ \text{-}\ t\ \Rightarrow T}$$

Give to us by assumption (the first premise). □