Hereditary Substitution for Stratified System F

Harley D. Eades III, Aaron Stump
Computer Science
The University of Iowa

Abstract

This paper proves normalization for Stratified System F, a type theory of predicative polymorphism studied by D. Leivant, by an extension of the method of hereditary substitution due to F. Pfenning. The advantage of normalization by hereditary substitution over normalization by reducibility is that the proof method is substantially less intricate, which promises to make it easier to apply to new theories.

1 Introduction

Girard's reducibility method is a well-known technique for proving normalization of simply typed and polymorphic lambda calculi. This technique is, however, very intricate, which makes it difficult to apply to new theories. Therefore an easier technique is of interest to the research community. Based on the work of Pfenning in [3, 4] and Prawitz in [5, 6] we propose a normalization proof technique using a notion of interpretation of types and hereditary substitution. The advantage of our proposal is that it is substantially less involved then normalization by reducibility and so may be easier to apply to new theories.

The first major part of the technique we are proposing is defining an interpretation of types. Interpretations of types are essentially sets of terms with a common type with respect to a context. We define interpretation of types on normal terms and then extend this definition to non-normal terms. A non-normal term is a member of the interpretation of a type if and only if it normalizes to a term in that interpretation. We use a slight modification from how Prawitz proposes the interpretation of types. He defines them on ground instances of open terms while we define ours on open terms directly [5, 6]. The definition for non-normal terms is very important, because if the typing rules are sound with respect to the interpretation of types then normalization is implied.

Now to prove the typing rules are sound with respect to the interpretation of types we need the main idea of hereditary substitution. This notion is used to show the interpretations are closed under substitutions. The main idea behind hereditary substitution is just like ordinary capture-avoiding substitution, except if a redex is formed as a result of the substitution then that redex is recursively reduced. More importantly, when this redex is reduced the type of the result of the reduction gets smaller with respect to a well-founded ordering on types. This insures that hereditary substitution is terminating. Using our proposed proof technique we prove normalization for a type theory of predicative polymorphism.

In [2] Leivant proposed such a type theory called Stratified System F, where the types are stratified into levels (or ranks) based on type-quantification. The types that belong to level zero have no type-quantification, the types at level one only quantify over types of level zero, and the types at level n quantify over the types of level $n-1$. Stratifying System F into levels effectively prevents impredicativity, where a type $\phi = \forall X.\phi'$ quantifies over types including itself. In Stratified System F, we can only have $\phi = \forall X^n.\phi'$ where X ranges over the set of types of level $n-1$ which does not include ϕ.

Leivant considers finite stratification in [2] where he proves that the set of numeric functions definable in Stratified System F is exactly $\mathscr{E}_4$ in the Grzegorcyk Hierarchy i.e. the super-elementary functions. Leivant and Danner extend stratified polymorphism to transfinite ordinals in [1] and prove that the numeric functions definable in a stratified polymorphic theory called $S2\lambda$ are exactly the primitive recursive functions.

$$
\begin{array}{lcl}
t & := & x \mid \lambda x : T.t \mid tt \mid \Lambda X : K.t \mid t[T] \\
T & := & X \mid T \to T \mid \forall X : K.T \\
K & := & *_0 \mid *_1 \mid \ldots
\end{array}
$$

Figure 1: Syntax of Terms, Types, and Kinds

$$
\begin{array}{lcl}
(\Lambda X : *_p.t)[\phi] & \rightsquigarrow & [\phi/X]t \\
(\lambda x : \phi.t)t' & \rightsquigarrow & [t'/x]t
\end{array}
$$

Figure 2: Reduction Rules

2 Stratified System F

We consider only the finite version of Stratified System F as proposed by Leivant in [2] with some slight modifications. One of the major differences of our version of Stratified System F compared to Leivant's is that we use a kinding relation which relates a level to a type with respect to some context, using algorithmic type/kind-checking rules. The syntax for Stratified System F can be found in Figure 1 and reduction rules in Figure 2. Both the kinding and typing relations depend on well-formed contexts (Figure 3).

As stated before we use kinds to denote the level of a type. We define algorithmic kind-checking rules in Figure 5. The context Γ, type ϕ, and kind $*_l$ are inputs and there are no outputs. The following lemma shows that all kindable types are kindable with respect to a well-formed context. Its proof and all other omitted proofs can be found in the Appendix.

Lemma 1. *If $\Gamma \vdash \phi : *_p$ then Γ Ok.*

The kind-checking rules in Figure 5 are almost identical to the leveling rules defined by Leivant, which are defined in Figure 4. There are two differences, one to the variable rule and the other to the forall-type rule. Determining the exact relationship must remain to future work, but we conjecture the systems are equivalent up to changing the levels for the type variables in terms.

$$
\frac{}{\cdot\ Ok} \qquad \frac{\Gamma\ Ok}{\Gamma, X : *_p\ Ok}
$$

$$
\frac{\Gamma \vdash \phi : *_p \quad \Gamma\ Ok}{\Gamma, x : \phi\ Ok}
$$

Figure 3: Well-formedness of Contexts

We define algorithmic type-checking rules in Figure 6. The context Γ and the term t are inputs while the type ϕ is an output. The type-checking rules depend on the kinding relation defined above. To insure substitutions over contexts behave in an expected manner, we rename variables as necessary to ensure contexts have at most one declaration per variable.

$$\frac{\Gamma \vdash_L \phi_1 : *_p \quad \Gamma \vdash_L \phi_2 : *_q}{\Gamma \vdash_L \phi_1 \to \phi_2 : *_{max(p,q)}} \qquad \frac{\Gamma, X : *_q \vdash_L \phi : *_p}{\Gamma \vdash_L \forall X : *_q.\phi : *_{max(p,q+1)}}$$

$$\frac{\Gamma(X) = *_p \quad p \leq q \quad \Gamma\ Ok}{\Gamma \vdash_L X : *_q}$$

Figure 4: Leivant's Stratified System F Kinding Rules

$$\frac{\Gamma \vdash \phi_1 : *_p \quad \Gamma \vdash \phi_2 : *_q}{\Gamma \vdash \phi_1 \to \phi_2 : *_{max(p,q)}} \qquad \frac{\Gamma, X : *_q \vdash \phi : *_p}{\Gamma \vdash \forall X : *_q.\phi : *_{max(p,q)+1}}$$

$$\frac{\Gamma(X) = *_p \quad p \leq q \quad \Gamma\ Ok}{\Gamma \vdash X : *_q}$$

Figure 5: Stratified System F Kinding Rules

3 The Interpretation of Types

We now define the interpretation $[\![\phi]\!]_\Gamma$ of types ϕ in typing context Γ. The interpretation formalizes a modified version of a recent philosophical suggestion of Prawitz, who seeks to give a constructivist theory of meaning for logical formulas, based on an idea of canonical proofs [5, 6]. The modification adopted here is to define the meaning of open terms directly, where Prawitz defines the meaning of open terms via the meaning of all their ground instances. The definition of the interpretation of types proceeds in two parts (here and in the cited works of Prawitz).

First, we define when a normal term is in the meaning of a type. The definition, given in Figure 7, is by recursion on the structure of the normal form and requires ϕ to be kindable. This definition is the restriction of the typing relation to normal terms. If ϕ is not kindable then the interpretation of ϕ is the empty set. This part of the definition contains both introduction and elimination forms. The thing to note here with this definition is introduction and elimination forms play equally important roles. Reducibility interpretations, in contrast, are defined in terms of elimination. The second part of the interpretation of types states that non-normal term t is in the interpretation of a type if and only if it has a normal form in the interpretation of the type: for non-normal term t, $t \in [\![\phi]\!]_\Gamma \Leftrightarrow \exists n.t \rightsquigarrow^! n \in [\![\phi]\!]_\Gamma$, where $t \rightsquigarrow^! t'$ is defined as $t \rightsquigarrow^* t'$ and t'

$$\frac{\Gamma(x) = \phi \quad \Gamma\ Ok}{\Gamma \vdash x : \phi} \qquad \frac{\Gamma, x : \phi_1 \vdash t : \phi_2}{\Gamma \vdash \lambda x : \phi_1.t : \phi_1 \to \phi_2}$$

$$\frac{\Gamma \vdash t_1 : \phi_1 \to \phi_2 \quad \Gamma \vdash t_2 : \phi_1}{\Gamma \vdash t_1 t_2 : \phi_2} \qquad \frac{\Gamma, X : *_p \vdash t : \phi}{\Gamma \vdash \Lambda X : *_p.t : \forall X : *_p.\phi}$$

$$\frac{\Gamma \vdash t : \forall X : *_l.\phi_1 \quad \Gamma \vdash \phi_2 : *_l}{\Gamma \vdash t[\phi_2] : [\phi_2/X]\phi_1}$$

Figure 6: Stratified System F Type-Checking Rules

is normal. The second part of the definition of interpretation of types is very important, because if we show that the type-checking rules are sound with respect to the interpretation of types; i.e., $\Gamma \vdash t : \phi \Rightarrow t \in [\![\phi]\!]_\Gamma$ then the definition implies that the type theory is normalizing.

$$
\begin{array}{lcl}
x \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \Gamma(x) = \phi \\
n_1 n_2 \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \exists \phi'.n_1 \in [\![\phi' \to \phi]\!]_\Gamma \wedge n_2 \in [\![\phi']\!]_\Gamma \\
\lambda x : \phi_1.n \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \exists \phi_2.\phi = \phi_1 \to \phi_2 \wedge n \in [\![\phi_2]\!]_{\Gamma,x:\phi_1} \\
\Lambda X : *_p.n \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \exists \phi'.\phi = \forall X : *_p.\phi' \wedge n \in [\![\phi']\!]_{\Gamma,X:*_p} \\
n[\phi'] \in [\![\phi]\!]_\Gamma & \Leftrightarrow & \exists \phi'', l.\phi = [\phi'/X]\phi'' \wedge \; \Gamma \vdash \phi' : *_l \wedge n \in [\![\forall X : *_l.\phi'']\!]_\Gamma
\end{array}
$$

Figure 7: Interpretation of Kindable Types for Normal Terms

4 Soundness of Typing

Before we can show soundness of our type-checking rules with respect to the interpretation of types we must define a well-founded ordering on types. We define an ordering $>_\Gamma$ on types in the following definition.

Definition 2. *The ordering $>_\Gamma$ is defined as the least relation satisfying the universal closures of the following formulas:*

$$
\begin{array}{lcl}
\phi_1 \to \phi_2 & >_\Gamma & \phi_1 \\
\phi_1 \to \phi_2 & >_\Gamma & \phi_2 \\
\forall X : *_l.\phi & >_\Gamma & [\phi'/X]\phi \textit{ where } \Gamma \vdash \phi' : *_l.
\end{array}
$$

We will refer to the reflexive closure of $>_\Gamma$ as $\geq_\Gamma$. The ordering on the arrow-type is simply the strict subexpression ordering, while the ordering on the forall-type is defined using its level. We know by the definition of the kinding relation that the level of a forall-type $\phi = \forall X.\phi'$ is strictly larger than the level of X and ϕ'. Thus, ϕ must be strictly larger than $[\phi''/X]\phi'$, because again by the definition of the kinding relation the level of ϕ'' is less than or equal to the level of X.

4.1 Well-foundness of ordering on types

Before we can prove that $>_\Gamma$ is indeed well-founded we must prove the following three lemmas. The first is level weakening for kinding which shows that if a type is kindable at some level l then that type can be kinded at strictly higher levels.

Lemma 3 (Level Weakening for Kinding)**.** *If $\Gamma \vdash \phi : *_r$ and $r < s$ then $\Gamma \vdash \phi : *_s$.*

The second result is substitution for kinding and context-ok. The first part of this result shows that each level is closed under substitutions. While the second part shows that any context Γ is closed under substitutions if Γ *Ok*. We define $[\phi/X]\Gamma$ for some type ϕ and context Γ as replacing every occurrence of X in the types of the variables in context Γ.

Lemma 4 (Substitution for Kinding,Context-Ok)**.** *Suppose $\Gamma \vdash \phi' : *_p$. If $\Gamma, X : *_p, \Gamma' \vdash \phi : *_q$ with a derivation of depth d, then $\Gamma, [\phi'/X]\Gamma' \vdash [\phi'/X]\phi : *_q$. Also, if $\Gamma, X : *_p, \Gamma'$ Ok with a derivation of depth d, then $\Gamma, [\phi'/X]\Gamma'$ Ok.*

The final result we have to show before being able to prove that $>_\Gamma$ is well-founded is type ordering. Type ordering says that if a type ϕ is in level p and there exists some strictly smaller type ϕ' with respect to $>_\Gamma$ then ϕ' should also be in level p.

Lemma 5 (Type Ordering)**.** *If* $\Gamma \vdash \phi : *_p$ *and* $\phi >_\Gamma \phi'$ *then* $\Gamma \vdash \phi' : *_p$.

Proof. This is a proof by case analysis on the kinding derivation of $\Gamma \vdash \phi : *_p$, with a case analysis on the derivation of $\phi >_\Gamma \phi'$.

Case.

$$\frac{\Gamma(X) = *_p \quad p \leq q \quad \Gamma\ Ok}{\Gamma \vdash X : *_q}$$

This case cannot arise, because we do not have $X >_\Gamma \phi$ for any type ϕ.

Case.

$$\frac{\Gamma \vdash \phi_1 : *_p \quad \Gamma \vdash \phi_2 : *_q}{\Gamma \vdash \phi_1 \to \phi_2 : *_{max(p,q)}}$$

By analysis of the derivation of the assumed ordering statement, we must have $\phi' \equiv \phi_1$ or $\phi' \equiv \phi_2$. If $\phi' \equiv \phi_1$ and $p \geq q$ then we have the required kind derivation for ϕ'. If $p < q$ then by level weakening $\Gamma \vdash \phi_1 : *_q$, and we have the required kinding derivation for ϕ'. The case for when $\phi' \equiv \phi_2$ is similar.

Case.

$$\frac{\Gamma, X : *_r \vdash \phi : *_s}{\Gamma \vdash \forall X : *_r.\phi : *_{max(r,s)+1}}$$

By analysis of the derivation of the assumed ordering statement, we must have $\phi' \equiv [\phi''/X]\phi$, for some type ϕ'' with $\Gamma \vdash \phi'' : *_r$. Let $t = max(r,s)+1$. Clearly, $s < t$, hence by level weakening $\Gamma, X : *_r \vdash \phi : *_t$ and by substitution for kinding $\Gamma \vdash [\phi''/X]\phi : *_t$, and we have the required kinding derivation for ϕ'.

□

We now have the desired results to prove that the ordering $>_\Gamma$ is well-founded.

Theorem 6. *The ordering* $>_\Gamma$ *is well-founded on types* ϕ *such that* $\Gamma \vdash \phi : *_l$ *for some* l.

Proof. Consider an arbitrary non-empty subset S of well-kinded types ϕ with respect to context Γ, where there exists a j such that $\Gamma \vdash \phi : *_j$ holds. We will prove that S has a minimal element in the $>_\Gamma$ ordering. Consider the set J defined to be $\{j \in \mathbb{N} \mid \exists \phi \in S.\Gamma \vdash \phi : *_j\}$. Of course, J has a minimal element, say $\hat{j}$. Consider an arbitrary $\phi \in S$ such that $\Gamma \vdash \phi : \hat{j}$. Such an element must exist, by the definition of J. If ϕ is minimal, we are done. So suppose ϕ is not minimal, and consider the set Q defined to be $\{\phi' \in S \mid \phi \geq_\Gamma \phi'\}$. Suppose there is an infinite sequence (ϕ_n) of types, with each type in Q. This sequence cannot be decreasing in the strict subexpression ordering, or it would be finite. So there must be some n such that $\phi_n \equiv \forall X : *_i.\phi'_n$ for some X, i, and ϕ'_n; and $\phi_{n+1} \equiv [\phi'/X]\phi'_n$, for some ϕ' with $\Gamma \vdash \phi' : *_i$. By induction on m and application of Type Ordering, we must have $\Gamma \vdash \phi_m : *_{\hat{j}}$ for each ϕ_m in the sequence. So we have $\Gamma \vdash \forall X : *_i.\phi'_n : *_{\hat{j}}$. By inversion on the kinding relation, we have $\Gamma, X : *_i \vdash \phi'_n : *_k$ for some $k < \hat{j}$. Now by the Substitution for Kinding, we must have $\Gamma \vdash \phi_{n+1} : *_k$, contradicting minimality of $\hat{j}$. Hence, the sequence (ϕ_n) is finite, and thus ends in a minimal element, as required. $\square$

4.2 Substitution for interpretation of types

Lemma 12 is used to show that the interpretations of types are closed under substitutions, and Theorem 13 heavily relies on this lemma. This is a very important result, because the crucial notion of hereditary substitution is embodied in its proof. The next lemma shows that all inhabited types are well-kinded. This allows one to reason about the type of a term without the fear that the type may not be kindable. This is needed in the proof of Theorem 13. While Lemma 8 and Lemma 9 are both used in the proofs of Lemma 12 and Theorem 13. Finally, Lemma 10 is only used in the proof of Lemma 12.

Lemma 7 (Regularity). *If $\Gamma \vdash t : \phi$ then $\Gamma \vdash \phi : *_p$ for some p.*

Lemma 8 (Context Weakening for Kinding). *If $\Gamma, \Gamma'', \Gamma'$ Ok, and $\Gamma, \Gamma' \vdash \phi : *_p$ then $\Gamma, \Gamma'', \Gamma' \vdash \phi : *_p$.*

Lemma 9 (Context Weakening for Interpretations of Types). *If $\Gamma, \Gamma', \Gamma''$ Ok and $n \in [\![\phi]\!]_{\Gamma,\Gamma''}$ then $n \in [\![\phi]\!]_{\Gamma,\Gamma',\Gamma''}$.*

Lemma 10 (Context Strengthening for Kinding,Context-Ok). *If $\Gamma, x : \phi', \Gamma' \vdash \phi : *_p$ with a proof derivation of depth d, then $\Gamma, \Gamma' \vdash \phi : *_p$. Also, if $\Gamma, x : \phi, \Gamma'$ Ok with a proof derivation of depth d, then Γ, Γ' Ok.*

Lemma 11 says that the interpretations of types are closed under type-variable substitution. This is used in the proofs of Lemma 12 and Theorem 13.

Lemma 11 (Type Substitution for the Interpretation of Types). *If $n \in [\![\phi']\!]_{\Gamma,X:*_l,\Gamma'}$ and $\Gamma \vdash \phi : *_l$ then $[\phi/X]n \in [\![[\phi/X]\phi']\!]_{\Gamma,[\phi/X]\Gamma'}$.*

Proof. This proof is by structural induction on n.

Case. n is a variable y. Clearly, $[\phi/X]n \equiv [\phi/X]y = y \in [\![\phi']\!]_{\Gamma,X:*_l,\Gamma'}$, and $(\Gamma, [\phi/X]\Gamma')(y) = [\phi/X]\phi'$. Also, we have $\Gamma, [\phi/X]\Gamma' \vdash [\phi/X]\phi' : *_p$ for some p, by Lemma 4. Hence, by the definition of the interpretation of types, $y \in [\![[\phi/X]\phi']\!]_{\Gamma,[\phi/X]\Gamma'}$.

Case. Let $n \equiv \lambda y : \psi.n'$. By the definition of the interpretation of types $\phi' \equiv \psi \to \psi'$. By the induction hypothesis $[\phi/X]n' \in [\![[\phi/X]\psi']\!]_{\Gamma,\Gamma',y:[\phi/X]\psi}$. Again by the definition of the interpretation of types $\lambda y : [\phi/X]\psi.[\phi/X]n' \equiv [\phi/X](\lambda y : \psi.n') \in [\![[\phi/X]\phi']\!]_{\Gamma,[\phi/X]\Gamma'}$.

Case. Let $n \equiv n_1 n_2$. By the definition of the interpretation of types $\phi' \equiv \psi$, $n_1 \in [\![\psi' \to \psi]\!]_{\Gamma,X:*_q,\Gamma'}$, and $n_2 \in [\![\psi']\!]_{\Gamma,X:*_q,\Gamma'}$. By the induction hypothesis $[\phi/X]n_1 \in [\![[\phi/X](\psi' \to \psi)]\!]_{\Gamma,[\phi/X]\Gamma'}$ and $[\phi/X]n_2 \in [\![[\phi/X]\psi']\!]_{\Gamma,[\phi/X]\Gamma'}$. Now by the definition of the interpretation of types $([\phi/X]n_1)([\phi/X]n_2) \in [\![[\phi/X]\psi]\!]_{\Gamma,[\phi/X]\Gamma'}$, since $[\phi/X]n_1$, cannot be a λ-abstraction.

Case. Let $n \equiv \Lambda Y : *_q.n'$. By the definition of the interpretation of types $\phi' = \forall Y : *_q.\psi$ and $n' \in [\![\psi]\!]_{\Gamma,X:*_l,\Gamma',Y:*_q}$. By the induction hypothesis $[\phi/X]n' \in [\![[\phi/X]\psi]\!]_{\Gamma,[\phi/X]\Gamma',Y:*_q}$ and by the definition of the interpretation of types $\Lambda Y : *_q.[\phi/X]n' \in [\![\forall Y : *_q.[\phi/X]\psi]\!]_{\Gamma,[\phi/X]\Gamma'}$ which is equivalent to $[\phi/X](\Lambda Y : *_q.n') \in [\![[\phi/X](\forall Y : *_q.\psi)]\!]_{\Gamma,[\phi/X]\Gamma'}$.

Case. Let $n \equiv n'[\psi]$. By the definition of the interpretation of types $\phi = [\psi/Y]\psi'$, for some Y, ψ, and there exists a q such that $\Gamma,X : *_l,\Gamma' \vdash \psi : *_q$, and $n' \in [\![\forall Y : *_q.\psi']\!]_{\Gamma,X:*_l,\Gamma'}$. By the induction hypothesis $[\phi/X]n' \in [\![[\phi/X](\forall Y : *_q.\psi')]\!]_{\Gamma,[\phi/X]\Gamma'}$. Therefore, by the definition of the interpretation of types $([\phi/X]n')[\psi] \in [\![[\psi/Y]([\phi/X]\psi')]\!]_{\Gamma,[\phi/X]\Gamma'}$, which is equivalent to $[\phi/X](n'[\psi]) \in [\![[\phi/X]([\psi/Y]\psi')]\!]_{\Gamma,[\phi/X]\Gamma'}$.

□

Lemma 12 (Substitution for the Interpretation of Types)**.** *If $n' \in [\![\phi']\!]_{\Gamma,x:\phi,\Gamma'}$, $n \in [\![\phi]\!]_{\Gamma}$, then $[n/x]n' \rightsquigarrow^! \hat{n} \in [\![\phi']\!]_{\Gamma,\Gamma'}$ and if n' is not a λ-abstraction or a Λ-abstraction and $\hat{n}$ is, then $\phi \geq_{\Gamma,\Gamma'} \phi'$.*

Proof. Throughout this proof we will refer to "if n' is not a λ-abstraction or a Λ-abstraction and $\hat{n}$ is then $\phi \geq_{\Gamma,\Gamma'} \phi'$" as **A**. We proceed by induction on a measure (ϕ,n') in the lexicographic combination of $>_{\Gamma,\Gamma'}$ and the strict subexpression ordering. We also show in detail why the induction hypothesis is applicable in each case, including showing why the types we will interpret are kindable. We case split on n' as follows:

By the definition of the interpretation of types $\Gamma \vdash \phi : *_p$ and $\Gamma,x : \phi,\Gamma' \vdash \phi' : *p$ for some p and q. Also, by Lemma 10, $\Gamma,\Gamma' \vdash \phi' : *_q$. We will use the previous facts throughout the rest of the proof without indication.

Case. n' is a variable.

Case. Suppose $n' \equiv x$. We must show $[n/x]x \rightsquigarrow^! \hat{n} \in [\![\phi']\!]_{\Gamma,\Gamma'}$ for some $\hat{n}$. Take n for $\hat{n}$. Then $[n/x]x \rightsquigarrow^! n \in [\![\phi]\!]_{\Gamma}$, and by Lemma 9, $n \in [\![\phi]\!]_{\Gamma,\Gamma'}$. By the definition of the interpretation of types $\phi \equiv \phi'$. Clearly, **A** is satisfied.

Case. Let $n' \equiv y \not\equiv x$. Then $[n/x]y = y$. Take y for $\hat{n}$. Clearly, $y \in [\![\phi']\!]_{\Gamma,\Gamma'}$. **A** is trivially satisfied.

Case. Let $n' \equiv \lambda y : \phi_1'.n''$. By assumption $n' \in [\![\phi']\!]_{\Gamma,x:\phi,\Gamma'}$. By the definition of the interpretation of types there exists a type ϕ_2', such that, $\phi' \equiv \phi_1' \to \phi_2'$ and $n'' \in [\![\phi_2']\!]_{\Gamma,x:\phi,\Gamma',y:\phi_1'}$. By inversion of the arrow-type kind-checking rule, $p = max(r,s)$, $\Gamma,\Gamma' \vdash \phi_1' : *_r$, and $\Gamma,\Gamma' \vdash \phi_2' : *_s$ for some r and s. By Lemma 8, $\Gamma,\Gamma',y : \phi_1' \vdash \phi_2 : *_s$, and by the lexicographic ordering we know, $(\phi',n') > (\phi',n'')$. Finally, we can apply the the induction hypothesis and obtain $[n/x]n'' \rightsquigarrow^! \hat{n} \in [\![\phi_2']\!]_{\Gamma,\Gamma',y:\phi_1'}$. By the definition of the interpretation of types, $\lambda y : \phi_1'.\hat{n} \in [\![\phi_1' \to \phi_2']\!]_{\Gamma,\Gamma'}$. Now $[n/x]n' \equiv [n/x]\lambda y : \phi_1'.n'' \equiv \lambda y : \phi_1'.[n/x]n'' \rightsquigarrow^! \lambda y : \phi_1'.\hat{n} \in [\![\phi']\!]_{\Gamma,\Gamma'}$. Hence, $\phi' \equiv \phi_1' \to \phi_2'$. Clearly, **A** is satisfied.

Case. Let $n' \equiv \Lambda X : *_l.n''$. By the definition of the interpretation of types, there exists a type ϕ'', such that, $\phi' \equiv \forall X : *_l.\phi''$. Also, $n'' \in [\![\phi'']\!]_{\Gamma,x:\phi,\Gamma',X:*_l}$. By inversion of the forall-type kind-checking rule, $p = max(l,s)+1$, and $\Gamma,\Gamma',X : *_l \vdash \phi'' : *_s$, for some s, and since $(\phi',n') > (\phi',n'')$, we can apply the induction hypothesis, hence, $[n/x]n'' \rightsquigarrow^! \hat{n} \in [\![\phi'']\!]_{\Gamma,\Gamma',X:*_l}$. By definition of the interpretation of types, $\Lambda X : *_l.\hat{n} \in [\![\forall X : *_l.\phi'']\!]_{\Gamma,\Gamma'}$. Now $[n/x]n' \equiv [n/x]\Lambda X : *_l.n'' \equiv \Lambda X : *_l.[n/x]n'' \rightsquigarrow^! \Lambda X : *_l.\hat{n} \in [\![\phi']\!]_{\Gamma,\Gamma'}$. Clearly, **A** is satisfied.

Case. Let $n' \equiv n''[\phi_1']$. By the definition of the interpretation of types, there exists a ϕ_2' and l, such that, $\phi' \equiv [\phi_1'/X]\phi_2'$, $n'' \in [\![\forall X : *_l.\phi_2']\!]_{\Gamma,x:\phi,\Gamma'}$, $\Gamma, x : \phi, \Gamma' \vdash \phi_1' : *_l$, and $\Gamma, x : \phi, \Gamma' \vdash \forall X : *_l.\phi_2' : *_s$, for some s. By Lemma 10, $\Gamma, \Gamma' \vdash \forall X : *_l.\phi_2' : *_s$. Now, $(\phi', n') > (\phi', n'')$, hence, by the induction hypothesis $[n/x]n'' \rightsquigarrow^! \hat{n}' \in [\![\forall X : *_l.\phi_2']\!]_{\Gamma,\Gamma'}$, and $\phi \geq_{\Gamma,\Gamma'} \forall X : *_l.\phi_2$. We do a case split on whether $\hat{n}'$ is a Λ-abstraction or not.

Case. If $\hat{n}' \not\equiv \Lambda X : *_l.\hat{n}''$ then $\hat{n}'[\phi_1'] \in [\![[\phi_1'/X]\phi_2']\!]_{\Gamma,\Gamma'}$ since $\hat{n}'[\phi_1']$ is normal. Take $\hat{n}'$ for $\hat{n}$ and **A** is trivially satisfied.

Case. If $\hat{n}' \equiv \Lambda X : *_l.\hat{n}''$ then $\hat{n}'[\phi_1'] \equiv (\Lambda X : *_l.\hat{n}'')[\phi_1'] \rightsquigarrow [\phi_1'/X]\hat{n}''$. Now $(\forall X : *_l.\phi_2', \hat{n}') > ([\phi_1'/X]\phi_2', [\phi_1'/X]\hat{n}'')$. By Lemma 11, $[\phi_1'/X]\hat{n}'' \rightsquigarrow q \in [\![[\phi_1'/X]\phi_2']\!]_{\Gamma,\Gamma'}$. Take q for $\hat{n}$. By the definition of our ordering on types $\phi \geq_\Gamma \forall X : *_l.\phi_2' >_{\Gamma,\Gamma'} [\phi_1'/X]\phi_2'$, thus, **A** holds.

Case. Let $n' \equiv n_1'n_2'$. By the definition of the interpretation of types there exists a type ϕ'', such that, $n_1' \in [\![\phi'' \to \phi']\!]_{\Gamma,x:\phi,\Gamma'}$, $n_2' \in [\![\phi'']\!]_{\Gamma,x:\phi,\Gamma'}$, and $\Gamma, x : \phi, \Gamma' \vdash \phi'' : *_r$ for some r. By Lemma 10, $\Gamma, \Gamma' \vdash \phi'' : *_r$. Applying the arrow-type kind-checking rule yields $\Gamma, \Gamma' \vdash \phi'' \to \phi'$ and by the lexicographic ordering, $(\phi', n_1'n_2') > (\phi', n_1')$ and $(\phi', n_1'n_2') > (\phi', n_2')$. Finally, by the induction hypothesis, $[n/x]n_1' \rightsquigarrow^! \hat{n}_1 \in [\![\phi'' \to \phi']\!]_{\Gamma,\Gamma'}$ and $[n/x]n_2' \rightsquigarrow^! \hat{n}_2 \in [\![\phi'']\!]_{\Gamma,\Gamma'}$.

Case. If $\hat{n}_1 \not\equiv \lambda y : \phi''.z$ then take $\hat{n}_1\hat{n}_2$ for $\hat{n}$. Now $[n/x]n' \equiv ([n/x]n_1')([n/x]n_2') \rightsquigarrow^! \hat{n}_1\hat{n}_2$. By the definition of the interpretation of types $\hat{n}_1\hat{n}_2 \in [\![\phi']\!]_{\Gamma,\Gamma'}$.

Case. Let $\hat{n}_1 \equiv \lambda y : \phi''.z$. We know $\phi \geq_\Gamma \phi'' \to \phi'$ and $\phi'' \to \phi' \geq_\Gamma \phi''$, hence $\phi \geq_\Gamma \phi''$, and $z \in [\![\phi']\!]_{\Gamma,\Gamma',y:\phi''}$. Now $[n/x]n' \rightsquigarrow^! \hat{n}_1\hat{n}_2 \equiv (\lambda y : \phi''.z)\hat{n}_2 \rightsquigarrow [\hat{n}_2/y]z$. Since $(\phi'' \to \phi', \hat{n}_1) > (\phi', [\hat{n}_2/y]z)$, by the induction hypothesis $[\hat{n}_2/y]z \rightsquigarrow \hat{z} \in [\![\phi']\!]_{\Gamma,\Gamma'}$. Take $\hat{z}$ for $\hat{n}$. We know, $\phi >_{\Gamma,\Gamma'} \phi'$, thus, **A** holds.

□

4.3 Concluding normalization

We are now ready to present our main result. The next theorem shows that the type-checking rules are sound with respect to the interpretation of types. By the definition of the interpretation of types the following result implies that Stratified System F is normalizing.

Theorem 13 (Type Soundness for the Interpretation of Types). *If* $\Gamma \vdash t : \phi$ *then* $t \in [\![\phi]\!]_\Gamma$.

Proof. This is a proof by induction on the structure of the typing derivation of t.

Case.

$$\frac{\Gamma(x) = \phi \quad \Gamma\ Ok}{\Gamma \vdash x : \phi}$$

By regularity $\Gamma \vdash \phi : *_l$ for some l, hence $[\![\phi]\!]_\Gamma$ is nonempty. Clearly, $x \in [\![\phi]\!]_\Gamma$ by the definition of the interpretation of types.

Case.

$$\frac{\Gamma, x : \phi_1 \vdash t : \phi_2}{\Gamma \vdash \lambda x : \phi_1.t : \phi_1 \to \phi_2}$$

By the induction hypothesis $t \in [\![\phi_2]\!]_{\Gamma, x:\phi_1}$. By the definition of the interpretation of types $t \rightsquigarrow^! n \in [\![\phi_2]\!]_{\Gamma, x:\phi_1}$. Again, by the definition of the interpretation of types $\lambda x : \phi_1.t \rightsquigarrow^! \lambda x : \phi_1.n \in [\![\phi_1 \to \phi_2]\!]_\Gamma$.

Case.

$$\frac{\Gamma \vdash t_1 : \phi_2 \to \phi_1 \quad \Gamma \vdash t_2 : \phi_2}{\Gamma \vdash t_1 t_2 : \phi_1}$$

By the induction hypothesis $t_1 \rightsquigarrow^! n_1 \in [\![\phi_2 \to \phi_1]\!]_\Gamma$, $t_2 \rightsquigarrow^! n_2 \in [\![\phi_2]\!]_\Gamma$, $\Gamma \vdash \phi_2 \to \phi_1 : *_p$, and $\Gamma \vdash \phi_2 : *_q$. By inversion of the arrow-type kind-checking rule, $\Gamma \vdash \phi_1 : *_r$, and by Lemma 8, $\Gamma, x : \phi_2, \Gamma' \vdash \phi_1 : *_r$. We do a case split on whether or not n_1 is a λ-abstraction. If not $n_1 n_2 \in [\![\phi_1]\!]_\Gamma$. Suppose $n_1 \equiv \lambda x : \phi_2.n_1'$. Then $t_1 t_2 \rightsquigarrow^* (\lambda x : \phi_2.n_1')n_2 \rightsquigarrow [n_2/x]n_1'$. By the definition of the interpretation of types $n_1' \in [\![\phi_1]\!]_{\Gamma, x:\phi_2}$ and by Lemma 9, $n_1' \in [\![\phi_1]\!]_{\Gamma, x:\phi_2, \Gamma'}$. Therefore, by Lemma 12 $[n_2/x]n_1' \rightsquigarrow^! \hat{n} \in [\![\phi_1]\!]_{\Gamma, \Gamma'}$.

Case.

$$\frac{\Gamma, X : *_p \vdash t : \phi}{\Gamma \vdash \Lambda X : *_p.t : \forall X : *_p.\phi}$$

By the induction hypothesis $t \in [\![\phi]\!]_{\Gamma, X:*_p}$. By definition of the interpretation of types, $t \rightsquigarrow^! n \in [\![\phi]\!]_{\Gamma, X:*_p}$. Again, by definition of the interpretation of types $\Lambda X : *_p.t \rightsquigarrow^! \Lambda X : *_p.n \in [\![\phi]\!]_\Gamma$.

Case.

$$\frac{\Gamma \vdash t : \forall X : *_l.\phi_1 \quad \Gamma \vdash \phi_2 : *_l}{\Gamma \vdash t[\phi_2] : [\phi_2/X]\phi_1}$$

By the induction hypothesis $t \in [\![\forall X : *_l.\phi_1]\!]_\Gamma$. By the definition of the interpretation of types $t \rightsquigarrow^! n \in [\![\forall X : *_l.\phi_1]\!]_\Gamma$. We do a case split on whether or not n is a Λ-abstraction. If not then again, by the definition of the interpretation of types $n[\phi_2] \in [\![[\phi_2/X]\phi_1]\!]_\Gamma$, therefore $t \in [\![[\phi_2/X]\phi_1]\!]_\Gamma$. Suppose $n \equiv \Lambda X : *_l.n'$. Then $t[\phi_2] \rightsquigarrow^* (\Lambda X : *_l.n')[\phi_2] \rightsquigarrow [\phi_2/X]n'$. By the definition of the interpretation of types $n' \in [\![\phi_1]\!]_{\Gamma, X:*_l}$, and by Lemma 9, $n' \in [\![\phi_1]\!]_{\Gamma, X:*_l, \Gamma'}$. Therefore, by Lemma 11 $[\phi_2/X]n' \in [\![[\phi_2/X]\phi_1]\!]_{\Gamma, [\phi_2/X]\Gamma'}$.

□

Interestingly, using the previous results type-preservation for normal forms is easy to show. Due to soundness, if a term t is typeable at some type ϕ, then it is a member of ϕ's interpretation, and by the definition of the interpretation of types, t's normal form is also a member of ϕ's interpretation. Finally, by Lemma 14, t's normal form is typeable at type ϕ. Thus, we obtain the next corollary.

Lemma 14. *If $n \in [\![\phi]\!]_\Gamma$ and Γ Ok then $\Gamma \vdash n : \phi$.*

Proof. This proof is by structural induction on n.

Case. Let n be some variable x. By the definition of the interpretation of types $\Gamma(x) = \phi$ and by applying the variable type-checking rule $\Gamma \vdash x : \phi$.

Case. Let $n \equiv \lambda x : \phi_1.n'$. By the definition of the interpretation of types there exists a ϕ_2 such that $\phi = \phi_1 \to \phi_2$ and $n' \in [\![\phi_2]\!]_{\Gamma,x:\phi_1}$. By the induction hypothesis $\Gamma, x : \phi_1 \vdash n' : \phi_2$ and by applying the λ-abstraction type-checking rule $\Gamma \vdash \lambda x : \phi_2.n' : \phi_1 \to \phi_2$.

Case. Let $n \equiv \Lambda X : *_p.n'$. By the definition of the interpretation of types there exists a ϕ' such that $\phi = \forall X : *_p.\phi'$ and $n' \in [\![\phi']\!]_{\Gamma,X:*_p}$. By the induction hypothesis $\Gamma, X : *_p \vdash n' : \phi'$, hence, by applying the Λ-abstraction type-checking rule $\Gamma \vdash \Lambda X : *_p.n' : \forall X : *_p.\phi'$.

Case. Let $n \equiv n_1 n_2$. By the definition of the interpretation of types there exists a ϕ' such that $n_1 \in [\![\phi' \to \phi]\!]_\Gamma$ and $n_2 \in [\![\phi']\!]_\Gamma$. By the induction hypothesis $\Gamma \vdash n_1 : \phi' \to \phi$ and $\Gamma \vdash n_2 : \phi'$. By applying the application type-checking rule $\Gamma \vdash n_1 n_2 : \phi$.

Case. Let $n \equiv n'[\phi']$. By the definition of the interpretation of types there exists a ϕ'' and l such that $\phi \equiv [\phi'/X]\phi''$, $\Gamma \vdash \phi' : *_l$, and $n' \in [\![\forall X : *_l.\phi'']\!]_\Gamma$. By the induction hypothesis $\Gamma \vdash n' : \forall X : *_l.\phi''$ and by applying the type-instantiation type-checking rule $\Gamma \vdash n'[\phi'] : [\phi'/X]\phi''$.

□

Corollary 15. *If $\Gamma \vdash t : \phi$ then there exists a n such that $t \rightsquigarrow^! n$ and $\Gamma \vdash n : \phi$.*

5 Hereditary Substitution Function

The proof of Lemma 12 could be constructively formalized into a function called the hereditary substitution function denoted $[t_2/x]^\phi t_1$ where terms t_1 and t_2, type ϕ, and free variable x are inputs and output term t' such that $t' = [t_2/x]^\phi t_1$. This function is defined in Figure 8. Please note, that in the type-instantiation case of the definition, if s_1 is indeed a Λ-abstraction then we only need to substitute ϕ' for X in s'_1 using ordinary capture-avoiding substitution, because when substituting a type for a type-variable there is no way to create a new redex. It is easy to see that this is indeed a terminating function.

Theorem 16. *The hereditary substitution function for Stratified System F is terminating.*

Proof. By straightforward induction on the lexicographic ordering (ϕ, t) with respect to our ordering on types $>_\Gamma$ and the subexpression ordering on terms. □

6 Conclusion

We have proposed a new proof technique using the notions of interpretation of types and hereditary substitution, to prove normalization of a predicative polymorphic type theory called Stratified System F. Finally, we saw the resulting algorithm we would obtain if we were to constructively formalize the proof of Lemma 12. In future work we hope to extend this proof method to higher ordinals, which hopefully will allow us to prove normalization for theories like Gödel's System T. We would also like to thank the anonymous PSTT reviewer for their helpful comments.

$$
\begin{array}{lcl}
[t/x]^{\phi}x & = & t \\
[t/x]^{\phi}y & = & y\text{, if } y \text{ is a variable distinct from } x. \\
[t/x]^{\phi}\lambda y:\phi'.t' & = & \lambda y:\phi'.([t/x]^{\phi}t') \\
[t/x]^{\phi}\Lambda X:*_l.t' & = & \Lambda X:*_l.([t/x]^{\phi}t') \\
[t/x]^{b}(t_1\ t_2) & = & ([t/x]^{b}t_1)([t/x]^{b}t_2) \\
[t/x]^{\phi\to\phi'}(t_1\ t_2) & = & \text{let } s_1 = ([t/x]^{\phi\to\phi'}t_1) \text{ in} \\
 & & \text{let } s_2 = ([t/x]^{\phi\to\phi'}t_2) \text{ in} \\
 & & \text{if } s_1 \equiv \lambda y:\phi.s'_1 \text{ for some } y \text{ and } s'_1 \text{ then} \\
 & & \quad [s_2/y]^{\phi}s'_1 \\
 & & \text{else} \\
 & & \quad (s_1\ s_2) \\
[t/x]^{b}t'[\phi'] & = & ([t/x]^{b}t')[\phi'] \\
[t/x]^{\forall X:*_l.\phi}t'[\phi'] & = & \text{let } s_1 = [t/x]^{\forall X:*_l.\phi}t' \text{ in} \\
 & & \text{if } s_1 \equiv \Lambda X:*_l.s'_1 \text{ for some } X,\ s'_1 \text{ and } \Gamma\vdash\phi':*_q \text{ such that } q\leq l \text{ then} \\
 & & \quad [\phi'/X]s'_1 \\
 & & \text{else} \\
 & & \quad s_1[\phi']
\end{array}
$$

Figure 8: Hereditary Substitution Function for Stratified System F

References

[1] N. Danner and D. Leivant. Stratified polymorphism and primitive recursion. *Mathematical. Structures in Comp. Sci.*, 9(4):507–522, 1999.

[2] D. Leivant. Finitely stratified polymorphism. *Inf. Comput.*, 93(1):93–113, 1991.

[3] F. Pfenning. On the undecidability of partial polymorphic type reconstruction. Technical report, Carnegie Mellon University, Pittsburgh, PA, USA, 1992.

[4] F. Pfenning. Structural cut elimination. In *LICS '95: Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science*, page 156, Washington, DC, USA, 1995. IEEE Computer Society.

[5] D. Prawitz. *Logical Consequence from a Constructivist Point of View*, pages 671–695. Volume 1 of Shapiro [7], 2005.

[6] D. Prawitz. Meaning approached via proofs. *Synthese*, 148(3):507–524, 2006.

[7] S. Shapiro. *The Oxford Handbook of Philosophy of Mathematics and Logic*. Oxford University Press, 2005.

[8] K. Watkins, I. Cervesato, F. Pfenning, and D. Walker. A concurrent logical framework: The propositional fragment. *Types for Proofs and Programs*, 3085:355–377, 2004.

A Proof of Lemma 1

This is a proof by structural induction on the kinding derivation of $\Gamma\vdash\phi:*_p$.

Case.

$$\frac{\Gamma(X) = *_p \quad p\leq q \quad \Gamma\ Ok}{\Gamma\vdash X:*_q}$$

By inversion of the kind-checking rule Γ *Ok*.

Case.

$$\frac{\Gamma \vdash \phi_1 : *_p \quad \Gamma \vdash \phi_2 : *_q}{\Gamma \vdash \phi_1 \to \phi_2 : *_{max(p,q)}}$$

By the induction hypothesis, $\Gamma \vdash \phi_1 : *_p$ and $\Gamma \vdash \phi_2 : *_q$ both imply Γ *Ok*. Since the arrow-type kind-checking rule does not modify Γ in anyway Γ will remain *Ok*.

Case.

$$\frac{\Gamma, X : *_q \vdash \phi : *_p}{\Gamma \vdash \forall X : *_q . \phi : *_{max(p,q)+1}}$$

By the induction hypothesis $\Gamma, X : *_p$ *Ok*, and by inversion of the type-variable well-formed contexts rule Γ *Ok*.

B Proof of Lemma 3 (Level Weakening For Kinding)

We show level weakening for kinding by structural induction on the kinding derivation of $\phi : *_r$.

Case.

$$\frac{\Gamma(X) = *_p \quad p \leq q \quad \Gamma\ Ok}{\Gamma \vdash X : *_q}$$

By assumption we know $q < s$, hence by reapplying the rule and transitivity we obtain $\Gamma \vdash X : *_s$.

Case.

$$\frac{\Gamma \vdash \phi_1 : *_p \quad \Gamma \vdash \phi_2 : *_q}{\Gamma \vdash \phi_1 \to \phi_2 : *_{max(p,q)}}$$

By the induction hypothesis $\Gamma \vdash \phi_1 : *_s$ and $\Gamma \vdash \phi_2 : *_s$ for some arbitrary $s > max(p,q)$. Therefore, by reapplying the rule we obtain $\Gamma \vdash \phi_1 \to \phi_2 : *_s$.

Case.

$$\frac{\Gamma, X : *_q \vdash \phi' : *_p}{\Gamma \vdash \forall X : *_q . \phi' : *_{max(p,q)+1}}$$

We know by assumption that $max(p,q) + 1 < s$ which implies that $max(p,q) < s - 1$. Now by the induction hypothesis $\Gamma, X : *_q \vdash \phi' : *_{s-1}$. Lastly, we reapply the rule and obtain $\Gamma \vdash \forall X : *_q . \phi' : *_s$.

C Proof of Lemma 4 (Substitution for Kinding,Context-Ok)

This is a prove by induction on d. We prove the first implication first, and then the second, doing a case analysis for each implication on the form of the derivation whose depth is being considered.

Case.

$$\frac{(\Gamma, X : *_p, \Gamma')(Y) = *_r \quad r \leq s \quad \Gamma, X : *_p, \Gamma' \; Ok}{\Gamma, X : *_p, \Gamma' \vdash Y : *_s}$$

By assumption $\Gamma \vdash \phi' : *_p$. We must consider whether or not $X \equiv Y$. If $X \equiv Y$ then $[\phi'/X]Y \equiv \phi'$, $r = p$, and $q = s$; this conclusion is equivalent to $\Gamma, [\phi'/X]\Gamma' \vdash \phi' : *_q$ and by the induction hypothesis $\Gamma, [\phi'/X]\Gamma' \; Ok$. If $X \not\equiv Y$ then $[\phi'/X]Y \equiv Y$ and by the induction hypothesis $\Gamma, [\phi'/X]\Gamma' \; Ok$, hence, $\Gamma, [\phi'/X]\Gamma' \vdash Y : *_q$.

Case.

$$\frac{\Gamma, X : *_p, \Gamma' \vdash \phi_1 : *_r \quad \Gamma, X : *_p, \Gamma' \vdash \phi_2 : *_s}{\Gamma, X : *_p, \Gamma' \vdash \phi_1 \to \phi_2 : *_{max(r,s)}}$$

Here $q = max(r,s)$ and by the induction hypothesis $\Gamma, [\phi'/X]\Gamma' \vdash [\phi'/X]\phi_1 : *_r$ and $\Gamma, [\phi'/X]\Gamma' \vdash [\phi'/X]\phi_2 : *_s$. We can now reapply the rule to get $\Gamma, [\phi'/X]\Gamma' \vdash [\phi'/X](\phi_1 \to \phi_2) : *_q$.

Case.

$$\frac{\Gamma, X : *_q, \Gamma', Y : *_r \vdash \phi : *_s}{\Gamma, X : *_p, \Gamma' \vdash \forall Y : *_r.\phi : *_{max(r,s)+1}}$$

Here $q = max(r,s)+1$ and by the induction hypothesis $\Gamma, [\phi'/X]\Gamma', Y : *_r \vdash [\phi'/X]\phi : *_s$. We can reapply this rule to get $\Gamma, [\phi'/X]\Gamma' \vdash [\phi'/X]\forall Y : *_r.\phi : *_q$.

We now show the second implication. The case were $d = 0$ cannot arise, since it requires the context to be empty. Suppose $d = n+1$. We do a case analysis on the last rule applied in the proof derivation of $\Gamma, X : *_p, \Gamma'$.

Case. Suppose $\Gamma' = \Gamma'', Y : *_q$.

$$\frac{\Gamma, X : *_p, \Gamma'' \; Ok}{\Gamma, X : *_p, \Gamma'', Y : *_q \; Ok}$$

By the induction hypothesis, $\Gamma, [\phi'/X]\Gamma'' \; Ok$. Now, by reapplying the rule above $\Gamma, [\phi'/X]\Gamma'', Y : *_q \; Ok$, hence $\Gamma, [\phi'/X]\Gamma' \; Ok$, since $X \not\equiv Y$.

Case. Suppose $\Gamma' = \Gamma'', y : \phi$.

$$\frac{\Gamma, X : *_p, \Gamma'' \vdash \phi : *_q \quad \Gamma, X : *_p, \Gamma''\ Ok}{\Gamma, X : *_p, \Gamma'', y : \phi\ Ok}$$

By the induction hypothesis, $\Gamma', [\phi'/X]\Gamma'' \vdash [\phi'/X]\phi : *_q$ and $\Gamma', [\phi'/X]\Gamma''\ Ok$. Thus, by reapplying the rule above $\Gamma, [\phi'/X]\Gamma'', x : [\phi'/X]\phi\ Ok$, therefore, $\Gamma, [\phi'/X]\Gamma'\ Ok$.

D Proof of Lemma 7 (Regularity)

This proof is by structural induction on the derivation of $\Gamma \vdash t : \phi$.

Case.

$$\frac{\Gamma(x) = \phi \quad \Gamma\ Ok}{\Gamma \vdash x : \phi}$$

By the definition of well-formedness contexts $\Gamma \vdash \phi : *_p$ for some p.

Case.

$$\frac{\Gamma, x : \phi_1 \vdash t : \phi_2}{\Gamma \vdash \lambda x : \phi_1.t : \phi_1 \to \phi_2}$$

By the induction hypothesis $\Gamma \vdash \phi_1 : *_p$ and $\Gamma \vdash \phi_1 : *_p$. By applying the arrow-type kind-checking rule we get $\Gamma \vdash \phi_1 \to \phi_2 : *_{max(p,q)}$.

Case.

$$\frac{\Gamma \vdash t_1 : \phi_1 \to \phi_2 \quad \Gamma \vdash t_2 : \phi_1}{\Gamma \vdash t_1 t_2 : \phi_2}$$

By the induction hypothesis $\Gamma \vdash \phi_1 \to \phi_2 : *r$ and $\Gamma \vdash \phi_1 : *_p$. By inversion of the arrow-type kind-checking rule $r = max(p,q)$, for some q, which implies $\Gamma \vdash \phi_2 : *_q$.

Case.

$$\frac{\Gamma, X : *_p \vdash t : \phi}{\Gamma \vdash \Lambda X : *_p.t : \forall X : *_q.\phi}$$

By the induction hypothesis $\Gamma, X : *_q \vdash \phi : *_p$. By applying the forall-type kind-checking rule $\Gamma \vdash \forall X.\phi : *_{max(p,q)+1}$.

Case.

$$\frac{\Gamma \vdash t : \forall X : *_p.\phi_1 \quad \Gamma \vdash \phi_2 : *_p}{\Gamma \vdash t[\phi_2] : [\phi_2/X]\phi_1}$$

By assumption $\Gamma \vdash \phi_2 : *_r$. By the induction hypothesis $\Gamma \vdash \forall X : *_p.\phi_1 : *_s$ and by inversion of the forall-type kind-checking rule $r = max(p,q)+1$, for some q, which implies $\Gamma, X : *_p \vdash \phi_1 : *_q$. Now, by Lemma 4, $\Gamma \vdash [\phi_2/X]\phi_1 : *_q$.

E Proof of Lemma 8 (Context Weakening For Kinding)

This is a proof by structural induction on the kinding derivation of $\Gamma, \Gamma' \vdash \phi : *_p$.

Case.

$$\frac{(\Gamma, \Gamma')(X) = *_p \quad p \leq q \quad \Gamma, \Gamma' \; Ok}{\Gamma, \Gamma' \vdash X : *_q}$$

If $(\Gamma, \Gamma')(X) = *_p$ then $(\Gamma, \Gamma'', \Gamma')(X) = *_p$, hence, by reapplying the type-variable kind-checking rule, $\Gamma, \Gamma'', \Gamma' \vdash \phi : *_p$.

Case.

$$\frac{\Gamma, \Gamma' \vdash \phi_1 : *_p \quad \Gamma, \Gamma' \vdash \phi_2 : *_q}{\Gamma, \Gamma' \vdash \phi_1 \to \phi_2 : *_{max(p,q)}}$$

By the induction hypothesis $\Gamma, \Gamma'', \Gamma' \vdash \phi_1 : *_p$ and $\Gamma, \Gamma'', \Gamma' \vdash \phi_2 : *_q$, hence, by reapplying the arrow-type kind-checking rule $\Gamma, \Gamma'', \Gamma'' \vdash \phi_1 \to \phi_2 : *_{max(p,q)}$.

Case.

$$\frac{\Gamma, \Gamma', X : *_q \vdash \phi' : *_p}{\Gamma, \Gamma' \vdash \forall X : *_q.\phi' : *_{max(p,q)+1}}$$

By the induction hypothesis $\Gamma, \Gamma'', \Gamma', X : *_p \vdash \phi : *_q$, hence, by reapplying the forall-type kind-checking rule $\Gamma, \Gamma'', \Gamma' \vdash \forall X : *_p.\phi : *_{max(p,q)+1}$.

F Proof of Lemma 9 (Context Weakening for Interpretations of Types)

This proof is by structural induction on n.

Case. Let $n \equiv x$. By the definition of the interpretation of types $\Gamma(x) = \phi$. Clearly, $(\Gamma, \Gamma')(x) = \phi$, and Lemma 8 gives us $\Gamma, \Gamma' \vdash \phi : *_p$ hence, $x \in [\![\phi]\!]_{\Gamma,\Gamma'}$.

Case. Let $n \equiv \lambda x : \phi_1.n'$. By the definition of the interpretation of types, there exists a type ϕ_2, such that $\phi = \phi_1 \to \phi_2$, and $n' \in [\![\phi_2]\!]_{\Gamma,x:\phi_1}$. By the induction hypothesis, $n' \in [\![\phi_2]\!]_{\Gamma,\Gamma',x:\phi_1,}$, and by the definition of the interpretation of types $\lambda x : \phi_1.n' \in [\![\phi_1 \to \phi_2]\!]_{\Gamma,\Gamma'}$.

Case. Let $n \equiv n_1 n_2$. By the definition of the interpretation of types, there exists a type ϕ_1, such that $n_1 \in [\![\phi_1 \to \phi_2]\!]_{\Gamma}$, and $n_2 \in [\![\phi_2]\!]_{\Gamma}$. By the induction hypothesis, $n_1 \in [\![\phi_1 \to \phi_2]\!]_{\Gamma,\Gamma'}$, and $n_2 \in [\![\phi_2]\!]_{\Gamma,\Gamma'}$. Thus, by the definition of the interpretation of types $n_1 n_2 \in [\![\phi_2]\!]_{\Gamma,\Gamma'}$.

Case. Let $n \equiv \Lambda X : *_p.n'$. By the definition of the interpretation of types, there exists a type ϕ', such that $n' \in [\![\phi']\!]_{\Gamma,X:*_p}$, and by the induction hypothesis $n' \in [\![\phi']\!]_{\Gamma,X:*_p,\Gamma'}$. By the definition of the interpretation of types $\Lambda X : *_p.n' \in [\![\forall X : *_p.\phi']\!]_{\Gamma,\Gamma'}$.

Case. Let $n \equiv n'[\phi']$. By the definition of the interpretation of types, there exists a type ϕ'' and l, such that $\phi = [\phi'/X]\phi''$, $\Gamma \vdash \phi' : *_l$, and $n' \in [\![\forall X : *_l.\phi'']\!]_{\Gamma}$. By the induction hypothesis $n' \in [\![\forall X : *_l.\phi'']\!]_{\Gamma,\Gamma'}$. We know, $\Gamma \vdash \phi' : *_k$, for some $k \leq l$, so by Lemma 8, $\Gamma,\Gamma' \vdash \phi' : *_k$, hence $\Gamma \vdash \phi' : *_l$. Thus, $n[\phi'] \in [\![[\phi'/X]\phi'']\!]_{\Gamma,\Gamma'}$.

G Proof of Lemma 10 (Context Strengthening for Kinding,Context-Ok)

This is a prove by induction on d. We prove the first implication first, and then the second, doing a case analysis for each implication on the form of the derivation whose depth is being considered.

Case.

$$\frac{(\Gamma,x:\phi',\Gamma')(X) = *_p \quad p \leq q \quad \Gamma,x:\phi',\Gamma'\ Ok}{\Gamma,x:\phi',\Gamma' \vdash X : *_q}$$

By the second implication of the induction hypothesis, $\Gamma,\Gamma'\ Ok$. Also, $(\Gamma,\Gamma')(X) = *_p$. Now by reapplying the rule above, $\Gamma,\Gamma' \vdash X : *_q$.

Case.

$$\frac{\Gamma,x:\phi',\Gamma' \vdash \phi_1 : *_p \quad \Gamma,x:\phi',\Gamma' \vdash \phi_2 : *_q}{\Gamma,x:\phi',\Gamma' \vdash \phi_1 \to \phi_2 : *_{max(p,q)}}$$

By the first implication of the induction hypothesis, $\Gamma,\Gamma' \vdash \phi_1 : *_p$ and $\Gamma,\Gamma' \vdash \phi_2 : *_q$. By reapplying the rule above we get, $\Gamma,\Gamma' \vdash \phi_1 \to \phi_2 : *_{max(p,q)}$.

Case.

$$\frac{\Gamma,x:\phi,\Gamma',Y:*_q \vdash \phi : *_p}{\Gamma,x:\phi',\Gamma' \vdash \forall Y : *_q.\phi : *_{max(p,q)+1}}$$

By the first implication of the induction hypothesis, $\Gamma,\Gamma',Y:*_q \vdash \phi : *_p$. By reapplying the rule we get, $\Gamma,\Gamma' \vdash \forall Y : *_q.\phi : *_{max(p,q)+1}$.

We now prove the second implication. The case where $d = 0$ cannot arise, since it requires the context to be empty. Suppose $d = n+1$. We do a case analysis on the last rule applied in the proof derivation of $\Gamma, x : \phi, \Gamma'$ *Ok*.

Case. Suppose $\Gamma' = \Gamma'', Y : *_l$. Then the last rule of the proof derivation of $\Gamma, x : \phi, \Gamma'$ *Ok* is as follows.

$$\frac{\Gamma, x : \phi, \Gamma'' \; Ok}{\Gamma, x : \phi, \Gamma'', Y : *_l \; Ok}$$

By the second implication of the induction hypothesis, Γ, Γ'' *Ok*. Now reapplying the rule we get, $\Gamma, \Gamma'', Y : *_l$ *Ok*, which is equivalent to Γ, Γ' *Ok*.

Case. Suppose $\Gamma' = \Gamma'', y : \phi'$. Then the last rule of the proof derivation of $\Gamma, x : \phi, \Gamma'$ *Ok* is as follows.

$$\frac{\Gamma, x : \phi, \Gamma'' \vdash \phi' : *_p \quad \Gamma, x : \phi, \Gamma'' \; Ok}{\Gamma, x : \phi, \Gamma'', y : \phi' \; Ok}$$

By the first implication of the induction hypothesis, $\Gamma, \Gamma'' \vdash \phi' : *_p$ and by the second, Γ, Γ'' *Ok*. Therefore, by reapplying the rule above, $\Gamma, \Gamma'', y : \phi'$ *Ok*, which is equivalent to Γ, Γ' *Ok*.