Typed Operational Semantics for Higher Order Subtyping

Adriana Compagnoni   Healfdene Goguen

abc@dcs.ed.ac.uk   hhg@dcs.ed.ac.uk

Department of Computer Science, University of Edinburgh
The King's Buildings, Edinburgh, EH9 3JZ, United Kingdom
Tel: (+44) (131) 650-1000   Fax: (+44) (131) 667-7209

Abstract

Bounded operator abstraction is a language construct relevant to object oriented programming languages and to ML2000, the successor to Standard ML. In this paper, we introduce $\mathcal{F}^{\omega}_{\leq}$, a variant of $F^{\omega}_{<:}$ with this feature and with Cardelli and Wegner's kernel Fun rule for quantifiers. We define a typed operational semantics with subtyping and prove that it is equivalent with $\mathcal{F}^{\omega}_{\leq}$, using a Kripke model to prove soundness. The typed operational semantics provides a powerful tool to establish the metatheoretic properties of $\mathcal{F}^{\omega}_{\leq}$, such as Church–Rosser, subject reduction, the admissibility of structural rules, and the equivalence with the algorithmic presentation of the system.

1 Introduction

During the last decade, object-oriented programming languages such as Smalltalk, C++, Modula 3, and Java have become popular because they encourage and facilitate software reuse and abstract design. In this time, the theoretical community has struggled to achieve a balance between safety and expressiveness of object-oriented programming languages, where safe languages use type systems to restrict the legal programs and thereby prevent errors, and expressive languages provide more constructs to allow the programmer to write programs more clearly or concisely.

A wide variety of language features has been proposed to model constructs from object-oriented programming languages in type systems, for example bounded quantification [21], recursive types [3], and matching [1, 9]. The feature we study in this paper is bounded abstraction on types, also called bounded operator abstraction. Cardelli and Harper are in favor of including this in ML2000 (private communication), the successor of Standard ML. The constructor is amply motivated by many examples due to Kim Bruce [6], including the following, which shows how bounded abstraction on types can be used to define the type of binary search trees with comparable elements:

$$\text{Comparable} = \Lambda X\!:\!\star\,.[\text{eq} : X \to X \to \text{Bool}, \text{lt} : X \to X \to \text{Bool}, \text{gt} : X \to X \to \text{Bool}]$$

$$\text{BinTree} = \Lambda X\!:\!\star\,.\Lambda Y\!\leq\!(\text{Comparable}\,X)\!:\!\star\,.[\begin{array}{lcl} \text{find} & : & Y \to \text{Bool}, \\ \text{insert} & : & Y \to \text{BinTree}, \\ \text{isEmpty} & : & \text{Bool}] \end{array}$$

The calculus we study in this paper, $\mathcal{F}^{\omega}_{\leq}$, is a minimal type theoretic presentation without all of the features needed to construct this example, such as object and recursive types. However, Abadi and Cardelli's study of the higher-order object calculus [2] has demonstrated that these object-oriented features do not present difficulties for the metatheory.

The focus of this paper is on the metatheoretic treatment of subtyping. We see the contributions of the paper as the following:

- We develop the metatheory of a particular type theory, $\mathcal{F}^{\omega}_{\leq}$, which captures important features for the foundations of object-oriented programming languages.
- We introduce a typed operational semantics for a language with subtyping, as an intermediate language for proving syntactic results about the type theory.
- We give a logical relation style interpretation of subtyping, which allows us to study properties of kinding and subtyping simultaneously. Although such a construction was essential in order to be able to use typed operational semantics, it could also form the basis for the study of a system with subtyping directly using an algorithm for subtyping.

The paper is structured as follows. In the remainder of the introduction we give background information to clarify the second and third points above. In Section 2 we introduce the syntax of $\mathcal{F}^{\omega}_{\leq}$. In Section 3 we introduce the typed operational semantics for this system. In Section 4 we develop the fundamental properties of types and kinds in $\mathcal{F}^{\omega}_{\leq}$. Section 5 gives the model construction that shows soundness of the typed operational semantics for the typing rules. In Section 6 uses the previous results to prove subject reduction for terms in $\mathcal{F}^{\omega}_{\leq}$. Section 7 shows the equivalence of the usual and algorithmic presentations of $\mathcal{F}^{\omega}_{\leq}$. Finally, in Section 8 we summarize related and future work, and Section 9 gives our conclusions.

1.1 Syntactic Properties of Interest

We believe that type-checking for programming languages should be decidable. Decidable type systems prevent basic programming errors by limiting the meaningful programs. While we want the type system to be powerful to allow more expressive programs, the type system should also have a low overhead for the programmer. In particular, the compiler should be able to recognize correct and incorrect programs reliably without help from the programmer. While the decidability of type- and subtype-checking for $\mathcal{F}^{\omega}_{\leq}$ are beyond the scope of this paper, the same technique used by Compagnoni [17] to prove these properties can be applied here.

For type systems with subtyping, an important aspect of decidable type-checking is the ability to eliminate instances of transitivity in subtyping. The transitivity rule leads to significant non-determinism, which in turn leads to infeasible subtyping algorithms. Thus existing algorithms for systems with decidable or semi-decidable subtyping [4, 16, 17, 32, 33] are syntax-directed in their search and only use transitivity in a specific, restricted way.

Another important property of a type system is subject reduction or type preservation, which states that evaluation of programs preserves their type. This is one of the central results of the paper. However, we also focus on the same property at the level of types, as well as showing strong normalization for types, which states that type reduction will always terminate. Both of these properties are needed to show the correctness of the algorithms for type-formation and subtyping.

1.2 Metatheory of Subtyping

Adding bounded operator abstraction to $F^{\omega}_{<:}$ leads to complications in studying metatheoretic properties of the system. The new constructor means that subtyping is now needed to check well-formation of types. Because type-checking is also needed in subtyping, this presents a circularity that together with β-equality is not trivial to study. In particular, we now need knowledge about subtyping to show results like subject reduction for types.

Most type systems with subtyping do not have this circularity: for example, $F^{\omega}_{<:}$ [11, 13, 12, 31, 14], $F^{\omega}_{\wedge}$ [18], and the systems in Abadi and Cardelli's book on objects [2] all separate the two judgements. Existing work on systems with such a circularity [4] avoids the interdependency by finding a particular order in which to prove results.

A similar interdependency exists in dependent type theory between the typing judgement and the equality of types. Typed operational semantics was originally developed for type theories with dependent types [24, 25], and gives a uniform treatment of syntactic properties such as substitution and generation lemmas, subject reduction, and strong normalization. By developing the metatheory of $\mathcal{F}^{\omega}_{\leq}$, this paper demonstrates that the technique can be extended successfully to type theories with subtyping. In particular, our approach of building a typed model for subtyping avoids many problems due to the circularity, by including extra structural information to allow proofs to go through.

Typed operational semantics also gives a clarification of proofs of strong normalization. The traditional proof of strong normalization for type theories builds a model where types are interpreted as sets of strongly normalizing terms. In this proof, details about strong normalization are mixed with the model: for example, such proofs rely on the fact that if e_1, e_2 and $e_1[x{\leftarrow}e_2]$ are strongly normalizing then $(\lambda x.e_1)(e_2)$ is strongly normalizing. In practice, such details are often suppressed, perhaps because they seem incongruous amidst the otherwise model-theoretic reasoning.

Typed operational semantics divides this proof into two conceptually different steps. First, we show results about well-formed terms in the typed operational semantics, such as that they are strongly normalizing. Secondly, we construct a model where types are interpreted as sets of terms of that type in the typed operational semantics. This gives us a type soundness result with respect to the operational semantics, informally similar to that for ML [36], although formally much stronger because the typed operational semantics encodes both type and reduction information. Composing the two results gives us strong normalization for the well-typed terms.

This describes the general approach of using typed operational semantics. However, because our goal is to study the subtyping relation, we build our construction over the language of types and kinds in $\mathcal{F}^{\omega}_{\leq}$, rather than over terms and types. Indeed, following our discussion above, while it should be possible to add non-terminating reductions to the language of terms, such as those for the object constructors of Abadi and Cardelli [2], it is our intention that the language of types and kinds should have desirable syntactic properties such as strong normalization. We can therefore use traditional approaches from type theory to study these properties.

Previous approaches to the metatheory of subtyping have used strong normalization of types as a basis for further reasoning about the subtyping relation. For example, Compagnoni [17] defines a system for subtyping normal types, and shows that this system is sound for rules of substitution and application, relying on a previous result that every well-formed type is strongly normalizing. In this paper, we instead extend the approach of typed operational semantics by building a logical relation style interpretation of the subtyping relation together with the interpretation of the typing relation. We are then able to study metatheoretic properties of well-formed types and the subtyping relation simultaneously, by reasoning in the typed operational semantics. Again, because we are concentrating on syntactic properties of subtyping, this does not follow the usual interpretation of subtyping in the literature as set inclusion [5, 7, 8, 10, 14, 18, 22, 32].

Our approach has the conceptual benefit of treating the interdependent judgements of kinding and subtyping simultaneously, which means that it is not sensitive to proving results in a specific order. Furthermore, the techniques we use here were originally developed for showing soundness of the semantics for the typing rules in dependent types. This suggests that our proof technique will be well-suited to studying more sophisticated type theories with subtyping, such as the Calculus of Constructions.

We now discuss the two steps of the model construction and the typed operational semantics in more detail, and mention which metatheoretic results follow from each step, with particular attention to the treatment of the subtyping judgement. We refer the reader to the original papers on typed operational semantics [24, 25, 26] for a more complete description of this technique.

1.3 The Typed Operational Semantics

The intermediate system in our proof, the typed operational semantics, offers an alternative induction principle to prove syntactic properties of type theories. We can use this system to prove more properties of types than simply strong normalization. Church–Rosser and subject reduction (Corollary 4.13) are particularly simple to show in the typed operational semantics, and therefore by soundness hold for the original typing system. We can also prove lemmas about replacing equal bounds and kinds in the context (Lemma 4.17) and transitivity elimination (Lemma 4.21) in the typed operational semantics. The power of this technique is still more evident in systems with η-equality [24], because Church–Rosser is only true for the well-typed terms, and therefore cannot be shown by purely syntactic means [30].

Finally, because the typed operational semantics is an algorithmic presentation of the type theory, we are able to use the equivalence of the typed operational semantics with the usual typing rules to prove the generation lemmas in Section 5.4 that are the basis for the metatheory of the term language of $\mathcal{F}^{\omega}_{\leq}$. This also allows us to prove in Section 7.6 the equivalence with the usual algorithmic presentation of the typechecking and subtyping relations, which include much less intermediate type information than typed operational semantics.

In our treatment, we have only given a typed operational semantics for the language of types and kinds, and the subtyping relation. This is because the full term language is intended to have recursion operators and objects, so the terms will not be strongly normalizing. The analysis of the language of types is still important, because it gives us information about the decomposition of subtyping judgements that allows us to prove subject reduction for terms and to show important properties about the typechecking and subtyping algorithms.

1.4 The Model Construction

The model construction for typed operational semantics is somewhat more complicated than the usual models for strong normalization proofs, because we build a model with well-formed types. We rely on several techniques used in the metatheory of dependent type theory:

- We build a Kripke-style model [20] with contexts as possible worlds: this is necessary to capture adding fresh variables to model the Π- and Λ-binders.
- We introduce a partial interpretation of kinds [34] to ensure that the kinding statement makes sense when modeling the kind $\Pi X{\leq}A{:}K_1.K_2$, which is only defined when the type A is well-formed.
- To give a typed treatment of subtyping we need to introduce a logical relation style interpretation of the subtyping judgement as well as the kinding judgement, based on a similar treatment of judgemental equality by Coquand [19].

We shall discuss the technical aspects of these constructions when we define the model in Section 5.

Although this may seem to complicate the proof considerably, these techniques are all well-established in the dependent type theory community. There are several justifications for preferring our approach. First, the extensions to the usual technique seem to be exactly what is required for the proof of soundness for the typed operational semantics. Secondly, we obtain an unexpected benefit by using the typed operational semantics and a model with kinded types: we are able to show the admissibility of the metatheoretic properties in Section 2.3, such as substitution, context replacement, and kind correctness, in the model construction, rather than showing them separately by induction on derivations. Finally, as we discuss in Section 8, using this model has suggested alternatives to the standard presentation of $F^{\omega}_{<:}$ that may satisfy transitivity elimination in the algorithm.

There is a simple intuition for why these results follow when we extend the model to kinded types. First, we notice that every proof of strong normalization needs to allow for substitution properties, because it is exactly this that allows us to model β-reduction. Hence, it is not surprising that rules like substitution are sound for what is essentially a model of strongly normalizing types with kind information.

Although we say that the model is built with well-kinded types, the types are well-kinded with respect to the typed operational semantics, a reduction sequence to normal form, not with respect to the kinding rules of $\mathcal{F}^{\omega}_{\leq}$. Because the reduction includes kinding information, it is possible to prove completeness: that a derivation of the well-formedness of a type in the typed operational semantics gives rise to a derivation of well-formedness in the usual typing system. However, the rules of inference for the typed operational semantics are restricted: there are rules for redices such as β, and structural rules allowing reduction within a type, but as usual for an algorithmic presentation, there are no general substitution rules. Hence, by appealing to soundness (Corollary 5.12), which eliminates uses of substitution in constructing a derivation in the typed operational semantics, and completeness (Proposition 4.6), which reflects the derivation without uses of substitution back into $\mathcal{F}^{\omega}_{\leq}$, we are able to eliminate all instances of the substitution rules.

We have therefore reduced the metatheory of a type theory to essentially two steps: first, develop some basic results of the system in the typed operational semantics, where syntactic results are relatively easy; and secondly, prove the equivalence of the typed operational semantics with the typing rules, where completeness is direct.

2 Syntax

We now present the rules for kinding, subtyping, and typing in $\mathcal{F}^{\omega}_{\leq}$. They are organized as proof systems for several interdependent judgement forms:

$\Gamma \vdash \mathrm{ok}$	well-formed context
$\Gamma \vdash K$	well-formed kind
$\Gamma \vdash K = K'$	kind equality
$\Gamma \vdash A : K$	well-kinded type
$\Gamma \vdash A = B : K$	type equality
$\Gamma \vdash A \leq B : K$	subtype
$\Gamma \vdash M : A$	well-typed term.

We sometimes use the metavariable J to range over statements (right-hand sides of judgements) of any of these judgement forms.

2.1 Syntactic Categories

The *kinds* of $\mathcal{F}^{\omega}_{\leq}$ are the kind $\star$ of proper types and the kinds $\Pi X{\leq}A{:}K_1.K_2$ of functions on types (sometimes called type operators).

$\mathbb{K}$	$::=$	$\star$	types
		$\Pi X{\leq}\mathbb{A}{:}\mathbb{K}.\mathbb{K}$	type operators

The language of *types* of $\mathcal{F}^{\omega}_{\leq}$ is a straightforward higher-order extension of $F_{\leq}$, Cardelli and Wegner's second-order calculus of bounded quantification. Like $F_{\leq}$, it includes type variables X; function types $A{\rightarrow}B$; and polymorphic types $\forall X{\leq}A{:}K.B$, in which the bound type variable X ranges over all subtypes of the upper bound A. Moreover, like F^{ω}, we allow types to be abstracted on types, but we allow bounds on the abstraction $\Lambda X{\leq}A{:}K.B$. We can also apply types to argument types $A\,B$; in effect, these forms introduce a simply typed λ-calculus with subtyping at the level of types. We shall sometimes use the word "types" to mean types and type operators.

The capture-avoiding substitution of A for X in B is written $B[X{\leftarrow}A]$. We identify types that differ only in the names of bound variables. We shall write $A(B_1, \ldots, B_n)$ for $((A\,B_1)\ldots B_n)$. If A is of the form $X(B_1, \ldots, B_n)$ then A has head variable X. We write $\mathrm{HV}(-)$ for the partial function returning the head variable of a term. We also extend the top type $\mathrm{T}_\star$ to any kind K by defining inductively $\mathrm{T}_{\Pi X \leq A:K_1.K_2} = \Lambda X{\leq}A{:}K_1.\mathrm{T}_{K_2}$.

$\mathbb{A}$	$::=$	X	type variable
		$\mathbb{A}{\rightarrow}\mathbb{A}$	function type
		$\forall X{\leq}\mathbb{A}{:}\mathbb{K}.\mathbb{A}$	polymorphic type
		$\Lambda X{\leq}\mathbb{A}{:}\mathbb{K}.\mathbb{A}$	operator abstraction
		$\mathbb{A}\,\mathbb{A}$	operator application
		$\mathrm{T}_\star$	top type

The language of terms includes the variables (x), applications ($M\ N$), and functional abstractions ($\lambda x{:}A.M$) of the simply typed λ-calculus, as well as bounded type abstraction ($\lambda X{\leq}A{:}K.M$) and application ($M\ A$) of F^ω. As in $F_\leq$, each type variable is given an upper bound at the point where it is introduced. We use the same notation for capture-avoiding substitution as that for types, and again identify α-equivalent terms.

$\mathbb{M}$	$::=$	x	variable
		$\lambda x{:}\mathbb{A}.\mathbb{M}$	abstraction
		$\mathbb{M}\mathbb{M}$	application
		$\lambda X{\leq}\mathbb{A}{:}\mathbb{K}.\mathbb{M}$	type abstraction
		$\mathbb{M}\mathbb{A}$	type application

The operational semantics of $\mathcal{F}^\omega_\leq$ is given by the following reduction rules on terms and types.

Definition 2.1 (*Untyped Reduction*)

1. $(\lambda x{:}A.e_1)e_2 \rightarrow_{\beta_1} e_1[x{\leftarrow}e_2]$
2. $(\lambda X{\leq}A{:}K_1.e)B \rightarrow_{\beta_1} e[X{\leftarrow}B]$
3. $(\Lambda X{\leq}A{:}K.B)C \rightarrow_{\beta_2} B[X{\leftarrow}C]$

Each relation ($\rightarrow_{\beta_1}$ and $\rightarrow_{\beta_2}$) is extended to a compatible relation with respect to term or type formation. The reduction $\rightarrow_\beta$ is defined by $\rightarrow_{\beta_1} \cup \rightarrow_{\beta_2}$. We write $\twoheadrightarrow_R$ for the transitive and reflexive closure of $\rightarrow_R$ and $=_R$ for the least equivalence relation containing $\rightarrow_R$. The β_2-normal form of a type A is written $\mathrm{nf}(A)$.

2.2 Contexts

A *context* Γ is a finite sequence of typing and subtyping assumptions for a set of term and type variables.

The empty context is written $\emptyset$. Term variable bindings have the form $x{:}A$; type variable bindings have the form $X{\leq}A{:}K$, where A is the upper bound of X and K is the kind of A.

Γ	$::=$	$\emptyset$	empty context
		$\Gamma,\ x{:}A$	term variable declaration
		$\Gamma,\ X{\leq}A{:}K$	type variable declaration

We call the set of term and type variables defined in a context Γ the *domain* of Γ, written $\mathrm{dom}(\Gamma)$. The functions $\mathrm{FV}(—)$ gives the set of free term variables and free type variables of a term, type, context, or statement. Since we are careful to ensure that no variable is bound more than once, we sometimes abuse notation and consider contexts as finite functions: $\Gamma(X)$ yields the bound of X in Γ, where X is implicitly asserted to be in $\mathrm{dom}(\Gamma)$.

We now give the rules of inference for the system $\mathcal{F}^\omega_\leq$.

2.3 Structural Rules

This section presents general structural rules for $\mathcal{F}^{\omega}_{\leq}$. In fact, each of the rules is admissible, which we shall show when we prove the equivalence of this system with the typed operational semantics.

$$\frac{\Gamma \vdash A : \star \qquad \Gamma \vdash J \qquad x \notin \mathrm{dom}(\Gamma)}{\Gamma,\, x{:}A \vdash J} \qquad (\textsc{Weak})$$

$$\frac{\Gamma \vdash J \qquad \Gamma \vdash A : K \qquad X \notin \mathrm{dom}(\Gamma)}{\Gamma,\, X{\leq}A{:}K \vdash J} \qquad (\textsc{TWeak})$$

$$\frac{\Gamma_1,\, X{\leq}B{:}K,\, \Gamma_2 \vdash J \qquad \Gamma_1 \vdash A{\leq}B : K}{\Gamma_1,\, \Gamma_2[X{\leftarrow}A] \vdash J[X{\leftarrow}A]} \qquad (\textsc{Subst})$$

$$\frac{\Gamma_1,\, x{:}A,\, \Gamma_2 \vdash J \qquad \Gamma_1 \vdash A =_\beta B : \star}{\Gamma_1,\, X{:}B,\, \Gamma_2 \vdash J} \qquad (\textsc{Context-EQ})$$

$$\frac{\Gamma_1,\, X{\leq}A{:}K,\, \Gamma_2 \vdash J \qquad \Gamma_1 \vdash A =_\beta B : K \qquad \Gamma_1 \vdash K =_\beta K'}{\Gamma_1,\, X{\leq}B{:}K',\, \Gamma_2 \vdash J} \qquad (\textsc{Context-T-EQ})$$

$$\frac{\Gamma \vdash B : K}{\Gamma \vdash K} \qquad (\textsc{Kind-Agreement})$$

2.4 Context Formation

The context formation rules are:

$$\emptyset \vdash \mathrm{ok} \qquad (\textsc{C-Empty})$$

$$\frac{\Gamma \vdash A : \star \qquad x \notin \mathrm{dom}(\Gamma)}{\Gamma,\, x{:}A \vdash \mathrm{ok}} \qquad (\textsc{C-Var})$$

$$\frac{\Gamma \vdash A : K \qquad X \notin \mathrm{dom}(\Gamma)}{\Gamma,\, X{\leq}A{:}K \vdash \mathrm{ok}} \qquad (\textsc{C-TVar})$$

2.5 Kind Formation

The well-formed kinds are those derived with the following rules.

$$\frac{\Gamma \ \vdash \mathrm{ok}}{\Gamma \vdash \star} \qquad (\text{K-}\star)$$

$$\frac{\Gamma,\, X{\leq}A{:}K_1 \vdash K_2}{\Gamma \vdash \Pi X{\leq}A{:}K_1.K_2} \qquad (\text{K-}\Pi)$$

2.6 Kind Equality

The interconvertibility of kinds is the propagation of the interconvertibility of types within kinds.

$$\frac{\Gamma \vdash K}{\Gamma \vdash K =_\beta K} \quad \text{(K-Eq-Refl)}$$

$$\frac{\Gamma \vdash K =_\beta K'}{\Gamma \vdash K' =_\beta K} \quad \text{(T-Eq-Sym)}$$

$$\frac{\Gamma \vdash K =_\beta K' \qquad \Gamma \vdash K' =_\beta K''}{\Gamma \vdash K =_\beta K''} \quad \text{(T-Eq-Trans)}$$

$$\frac{\Gamma \ \vdash \mathrm{ok}}{\Gamma \vdash \star =_\beta \star} \quad \text{(K-Eq-}\star\text{)}$$

$$\frac{\Gamma \vdash K_1 =_\beta K_1' \qquad \Gamma \vdash A =_\beta A' : K_1 \qquad \Gamma,\, X{\le}A{:}K_1 \vdash K_2 =_\beta K_2'}{\Gamma \vdash \Pi X{\le}A{:}K_1.K_2 =_\beta \Pi X{\le}A'{:}K_1'.K_2'} \quad \text{(K-Eq-}\Pi\text{)}$$

2.7 Type Formation

For each type constructor, we give a rule specifying how it can be used to build well-formed type expressions. The new rules for type formation are the ones that deal with bounded type abstraction (T-TAbs), type application (T-TApp), and kind conversion (T-Conv).

$$\frac{\Gamma \ \vdash \mathrm{ok}}{\Gamma \vdash \mathrm{T}_\star : \star} \quad \text{(T-Top)}$$

$$\frac{\Gamma_1,\, X{\le}A{:}K,\, \Gamma_2 \vdash \mathrm{ok}}{\Gamma_1,\, X{\le}A{:}K,\, \Gamma_2 \vdash X : K} \quad \text{(T-TVar)}$$

$$\frac{\Gamma \vdash A_1 : \star \qquad \Gamma \vdash A_2 : \star}{\Gamma \vdash A_1{\to}A_2 : \star} \quad \text{(T-Arrow)}$$

$$\frac{\Gamma,\, X{\le}A_1{:}K \vdash A_2 : \star}{\Gamma \vdash \forall X{\le}A_1{:}K.A_2 : \star} \quad \text{(T-All)}$$

$$\frac{\Gamma,\, X{\le}A_1{:}K_1 \vdash A_2 : K_2}{\Gamma \vdash \Lambda X{\le}A_1{:}K_1.A_2 : \Pi X{\le}A_1{:}K_1.K_2} \quad \text{(T-TAbs)}$$

$$\frac{\Gamma \vdash A : \Pi X{\le}B{:}K_1.K_2 \qquad \Gamma \vdash C{\le}B : K_1}{\Gamma \vdash A\,C : K_2[X{\leftarrow}C]} \quad \text{(T-TApp)}$$

$$\frac{\Gamma \vdash A : K \qquad \Gamma \vdash K =_\beta K'}{\Gamma \vdash A : K'} \quad \text{(T-Conv)}$$

2.8 Type Equality

The judgemental type equality is generated by the typed beta-equality rule (T-Eq-Beta). It is a congruence with respect to type formation and incorporates kind equivalence so that equal kinds contain the same equality relation on types.

$$\frac{\Gamma,\, X{\le}A_1{:}K_1 \vdash A_2 : K_2 \qquad \Gamma \vdash C \le A_1 : K_1}{\Gamma \vdash (\Lambda X{\le}A_1{:}K_1.A_2)\, C =_\beta A_2[X{\leftarrow}C] : K_2[X{\leftarrow}C]} \quad \text{(T-Eq-Beta)}$$

$$\frac{\Gamma \vdash A : K}{\Gamma \vdash A =_\beta A : K} \quad \text{(T-Eq-Refl)}$$

$$\frac{\Gamma \vdash A =_\beta B : K}{\Gamma \vdash B =_\beta A : K} \quad \text{(T-Eq-Sym)}$$

$$\frac{\Gamma \vdash A =_\beta B : K \qquad \Gamma \vdash B =_\beta C : K}{\Gamma \vdash A =_\beta C : K} \quad \text{(T-Eq-Trans)}$$

$$\frac{\Gamma \vdash A_1 =_\beta B_1 : \star \qquad \Gamma \vdash A_2 =_\beta B_2 : \star}{\Gamma \vdash A_1 \rightarrow A_2 =_\beta B_1 \rightarrow B_2 : \star} \quad \text{(T-Eq-Arrow)}$$

$$\frac{\Gamma \vdash A_1 =_\beta B_1 : K \qquad \Gamma, X \leq A_1 {:} K \vdash A_2 =_\beta B_2 : \star \qquad \Gamma \vdash K =_\beta K'}{\Gamma \vdash \forall X \leq A_1 {:} K . A_2 =_\beta \forall X \leq B_1 {:} K' . B_2 : \star} \quad \text{(T-Eq-All)}$$

$$\frac{\Gamma \vdash A_1 =_\beta B_1 : K_1 \qquad \Gamma, X \leq A_1 {:} K_1 \vdash A_2 =_\beta B_2 : K_2 \qquad \Gamma \vdash K_1 =_\beta K_1'}{\Gamma \vdash \Lambda X \leq A_1 {:} K_1 . A_2 =_\beta \Lambda X \leq B_1 {:} K_1' . B_2 : \Pi X \leq A_1 {:} K_1 . K_2} \quad \text{(T-Eq-TAbs)}$$

$$\frac{\Gamma \vdash A =_\beta B : \Pi X \leq E {:} K_1 . K_2 \qquad \Gamma \vdash C =_\beta D : K_1 \qquad \Gamma \vdash C \leq E : K_1}{\Gamma \vdash A\, C =_\beta B\, D : K_2[X \leftarrow C]} \quad \text{(T-Eq-TApp)}$$

$$\frac{\Gamma \vdash A =_\beta B : K \qquad \Gamma \vdash K =_\beta K'}{\Gamma \vdash A =_\beta B : K'} \quad \text{(T-Eq-Conv)}$$

2.9 Subtyping

The subtyping rules are those of $F^\omega_{<:}$ [11, 13, 12, 31, 14], except for those dealing with bounded type abstraction and type application shown below and the rule for subtyping the quantifier. We chose the Cardelli and Wegner's kernel Fun rule for quantifiers with equal bounds [15]. The contravariant rule for quantifiers renders the system undecidable, and transitivity elimination in the presence of such a rule in the higher-order case remains an open problem.

$$\frac{\Gamma \vdash A : K}{\Gamma \vdash A \leq \mathrm{T}_K : K} \quad \text{(S-Top)}$$

$$\frac{\Gamma \vdash A =_\beta B : K}{\Gamma \vdash A \leq B : K} \quad \text{(S-Conv)}$$

$$\frac{\Gamma \vdash A \leq B : K \qquad \Gamma \vdash B \leq C : K}{\Gamma \vdash A \leq C : K} \quad \text{(S-Trans)}$$

$$\frac{\Gamma_1, X \leq A {:} K, \Gamma_2 \vdash \mathrm{ok}}{\Gamma_1, X \leq A {:} K, \Gamma_2 \vdash X \leq A : K} \quad \text{(S-TVar)}$$

$$\frac{\Gamma \vdash B_1 \leq A_1 : \star \qquad \Gamma \vdash A_2 \leq B_2 : \star}{\Gamma \vdash A_1 \rightarrow A_2 \leq B_1 \rightarrow B_2 : \star} \quad \text{(S-Arrow)}$$

$$\frac{\Gamma, X \leq C {:} K \vdash A \leq B : \star}{\Gamma \vdash \forall X \leq C {:} K . A \leq \forall X \leq C {:} K . B : \star} \quad \text{(S-All)}$$

$$\frac{\Gamma,\, X{\leq}C{:}K_1 \vdash A \leq B : K_2}{\Gamma \vdash \Lambda X{\leq}C{:}K_1.A \leq \Lambda X{\leq}C{:}K_1.B : \Pi X{\leq}C{:}K_1.K_2} \qquad \text{(S-TAbs)}$$

$$\frac{\Gamma \vdash A \leq B : \Pi X{\leq}D{:}K_1.K_2 \qquad \Gamma \vdash C \leq D : K_1}{\Gamma \vdash A\,C \leq B\,C : K_2[X{\leftarrow}C]} \qquad \text{(S-TApp)}$$

$$\frac{\Gamma \vdash A \leq B : K \qquad \Gamma \vdash K =_\beta K'}{\Gamma \vdash A \leq B : K'} \qquad \text{(S-K-Conv)}$$

2.10 Term Formation

The term formation rules are those of the second-order calculus of bounded quantification with the difference that we include kind annotations in terms, types, contexts, and subtyping judgements.

$$\frac{\Gamma_1,\, x{:}A,\, \Gamma_2 \vdash \mathrm{ok}}{\Gamma_1,\, x{:}A,\, \Gamma_2 \vdash x : A} \qquad \text{(T-Var)}$$

$$\frac{\Gamma,\, x{:}A \vdash M : B}{\Gamma \vdash \lambda x{:}A.M : A {\to} B} \qquad \text{(T-Abs)}$$

$$\frac{\Gamma \vdash M : A {\to} B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B} \qquad \text{(T-App)}$$

$$\frac{\Gamma,\, X{\leq}A{:}K \vdash M : B}{\Gamma \vdash \lambda X{\leq}A{:}K.M : \forall X{\leq}A{:}K.B} \qquad \text{(T-TAbs)}$$

$$\frac{\Gamma \vdash M : \forall X{\leq}A{:}K.B \qquad \Gamma \vdash C \leq A : K}{\Gamma \vdash M\,C : B[X{\leftarrow}C]} \qquad \text{(T-TApp)}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash A \leq B : \star}{\Gamma \vdash M : B} \qquad \text{(T-Sub)}$$

3 The Typed Operational Semantics

The typed operational semantics for $\mathcal{F}^\omega_\leq$ is organized in five judgement forms.

$\Gamma \vdash_S \mathrm{ok}$	well-formed context
$\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$	kind normalization
$\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}{\twoheadrightarrow_{\mathrm{n}}}\; C : K$	type reduction
$\Gamma \vdash_S A \leq_W B : K$	weak-head subtyping
$\Gamma \vdash_S A \leq B : K$	subtyping

The informal meaning of these judgements is as follows. In $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$, K' is the normal form of K. In $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}{\twoheadrightarrow_{\mathrm{n}}}\; C : K$, B is the weak head normal form of A and C its normal form. In $\Gamma \vdash_S A \leq_W B : K$, A and B are in weak head normal form, and in $\Gamma \vdash_S A \leq B : K$, A and B are arbitrary types or type operators.

Definition 3.1 (*Weak-Head Normal*)
$\mathrm{T}_\star$, $A_1 \to A_2$, $\forall X{\leq}A{:}K.B$, and $\Lambda X{\leq}A{:}K.B$ are weak head normal.
$X(A_1, \ldots, A_n)$ is weak head normal if $A_1, \ldots, A_n$ are in normal form.

In order to prove the admissibility of transitivity in the semantics, we need to consider a stronger definition of weak head normal form. We consider expressions of the form $X(A_1, \ldots, A_n)$ weak head normal only if each A_i is fully normalized. It may be possible to strengthen the model in Section 5 and use the standard definition of this notion instead.

We use the following notations:

- $\Gamma \vdash_S A : K$ is notation for $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K$, for some B, C.
- $\Gamma \vdash_S K$ is notation for $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$, for some K'.
- $\Gamma \vdash_S A \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} B : K$ is notation for $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} A \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} B : K$.
- $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B : K$ is notation for $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K$, for some C.
- $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{n}} B : K$ means $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} C \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} B : K$, for some C.
- $\Gamma \vdash_S A, B \twoheadrightarrow_{\mathrm{n}} C : K$ means $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{n}} C : K$ and $\Gamma \vdash_S B \twoheadrightarrow_{\mathrm{n}} C : K$.
- $\Gamma \vdash_S K, K' \twoheadrightarrow_{\mathrm{n}} K''$ means $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K''$ and $\Gamma \vdash_S K' \twoheadrightarrow_{\mathrm{n}} K''$.

The rules are presented as simultaneously defined inductive relations.

3.1 Context Formation

$$\emptyset \vdash_S \mathrm{ok} \qquad \text{(SC-Empty)}$$

$$\frac{\Gamma \vdash_S A : \star \qquad x \notin \mathrm{dom}(\Gamma)}{\Gamma,\, x{:}A \vdash_S \mathrm{ok}} \qquad \text{(SC-Var)}$$

$$\frac{\Gamma \vdash_S A : K' \qquad \Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K' \qquad X \notin \mathrm{dom}(\Gamma)}{\Gamma,\, X{\leq}A{:}K \vdash_S \mathrm{ok}} \qquad \text{(SC-TVar)}$$

3.2 Kind Normalization

$$\frac{\Gamma \vdash_S \mathrm{ok}}{\Gamma \vdash_S \star \twoheadrightarrow_{\mathrm{n}} \star} \qquad \text{(SK-}\star\text{)}$$

$$\frac{\Gamma \vdash_S K_1 \twoheadrightarrow_{\mathrm{n}} K_1' \qquad \Gamma \vdash_S A \twoheadrightarrow_{\mathrm{n}} B : K_1' \qquad \Gamma,\, X{\leq}A : K_1 \vdash_S K_2 \twoheadrightarrow_{\mathrm{n}} K_2'}{\Gamma \vdash_S \Pi X{\leq}A{:}K_1.K_2 \twoheadrightarrow_{\mathrm{n}} \Pi X{\leq}B{:}K_1'.K_2'} \qquad \text{(SK-}\Pi\text{)}$$

Context formation and kind normalization rules follow from modifications to the context formation and kind equality rules of the system in Section 2.2. For example, in the type variable rule (ST-TVar) the kind of A and the kind in the declaration of X are β-equal but not necessarily identical.

3.3 Type Reduction

$$\frac{\Gamma \vdash_S \text{ok}}{\Gamma \vdash_S \mathrm{T}_\star \twoheadrightarrow_{\mathrm{w}} \mathrm{T}_\star \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} \mathrm{T}_\star : \star} \quad \text{(ST-Top)}$$

$$\frac{\Gamma \vdash_S A : K' \qquad \Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K' \qquad (X{\leq}A : K) \in \Gamma}{\Gamma \vdash_S X \twoheadrightarrow_{\mathrm{w}} X \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} X : K'} \quad \text{(ST-TVar)}$$

$$\frac{\begin{array}{ll}\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} D : \Pi X{\leq}C{:}K_1.K_2 & \Gamma \vdash_S E \leq_W C : K_1 \\ \Gamma \vdash_S B \twoheadrightarrow_{\mathrm{w}} E \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} F : K_1 & \Gamma \vdash_S K_2[X{\leftarrow}B] \twoheadrightarrow_{\mathrm{n}} K \quad D \neq \Lambda Y{\leq}G{:}K_3.H\end{array}}{\Gamma \vdash_S A\,B \twoheadrightarrow_{\mathrm{w}} D\,F \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} D\,F : K} \quad \text{(ST-TApp)}$$

$$\frac{\Gamma \vdash_S A_1 \twoheadrightarrow_{\mathrm{n}} B_1 : \star \qquad \Gamma \vdash_S A_2 \twoheadrightarrow_{\mathrm{n}} B_2 : \star}{\Gamma \vdash_S (A_1{\to}A_2) \twoheadrightarrow_{\mathrm{w}} (A_1{\to}A_2) \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} (B_1{\to}B_2) : \star} \quad \text{(ST-Arrow)}$$

$$\frac{\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{n}} C : K' \qquad \Gamma, X{\leq}A{:}K \vdash_S B \twoheadrightarrow_{\mathrm{n}} D : \star \qquad \Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'}{\Gamma \vdash_S \forall X{\leq}A{:}K.B \twoheadrightarrow_{\mathrm{w}} \forall X{\leq}A{:}K.B \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} \forall X{\leq}C{:}K'.D : \star} \quad \text{(ST-All)}$$

$$\frac{\Gamma \vdash_S K_1 \twoheadrightarrow_{\mathrm{n}} K_1' \qquad \Gamma \vdash_S A \twoheadrightarrow_{\mathrm{n}} C : K_1' \qquad \Gamma, X{\leq}A{:}K_1 \vdash_S B \twoheadrightarrow_{\mathrm{n}} D : K_2}{\Gamma \vdash_S \Lambda X{\leq}A{:}K_1.B \twoheadrightarrow_{\mathrm{w}} \Lambda X{\leq}A{:}K_1.B \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} \Lambda X{\leq}C{:}K_1'.D : \Pi X{\leq}C{:}K_1'.K_2} \quad \text{(ST-TAbs)}$$

$$\frac{\begin{array}{ll}\Gamma \vdash_S B \twoheadrightarrow_{\mathrm{w}} \Lambda X{\leq}A{:}K_1.D : \Pi X{\leq}A'{:}K_1'.K_2 & \Gamma \vdash_S K_2[X{\leftarrow}C] \twoheadrightarrow_{\mathrm{n}} K \\ \Gamma \vdash_S D[X{\leftarrow}C] \twoheadrightarrow_{\mathrm{w}} E \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} F : K & \Gamma \vdash_S C \leq A : {K_1}'\end{array}}{\Gamma \vdash_S B\,C \twoheadrightarrow_{\mathrm{w}} E \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} F : K} \quad \text{(ST-Beta)}$$

The rules for type reduction combine kinding information and computational behavior in the form of weak head and β-normal forms. For example, the rule for arrow types says how to obtain the weak head and β-normal form of $(A_1{\to}A_2)$ in $\star$ knowing those for A_1 and A_2 in $\star$ as well.

The beta rule, besides uncovering the outermost redex of the application $B\,C$, and contracting it, finds the weak head normal form E and the normal form F. The premise $\Gamma \vdash_S K_2[X{\leftarrow}C] \twoheadrightarrow_{\mathrm{w}} K$ ensures that E and F have β-equal kinds, and the subtyping premise $\Gamma \vdash_S C \leq A : {K_1}'$ enforces the well-formation of $B\,C$.

The subtyping relation is defined using two judgements: one deals with types in weak head normal form ($\Gamma \vdash_S A \leq_W B : K$) and the other with arbitrary types ($\Gamma \vdash_S A \leq B : K$).

3.4 Weak Head Subtyping

$$\frac{\Gamma \vdash_S A \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} B : \star \qquad \mathrm{HV}(A) \text{ undefined}}{\Gamma \vdash_S A \leq_W \mathrm{T}_\star : \star} \quad \text{(SWS-Top)}$$

$$\frac{\begin{array}{ll}\Gamma \vdash_S X(A_1,\ldots,A_m) \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K & \\ \Gamma \vdash_S \Gamma(X) \twoheadrightarrow_{\mathrm{n}} B : K' & \Gamma \vdash_S E \leq_W A : K \\ \Gamma \vdash_S B(A_1,\ldots,A_m) \twoheadrightarrow_{\mathrm{w}} E : K & A \neq X(A_1,\ldots,A_m)\end{array}}{\Gamma \vdash_S X(A_1,\ldots,A_m) \leq_W A : K} \quad \text{(SWS-TApp)}$$

$$\frac{\Gamma \vdash_S X(A_1, \ldots, A_m) \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} B : K}{\Gamma \vdash_S X(A_1, \ldots, A_m) \leq_W X(A_1, \ldots, A_m) : K} \quad \text{(SWS-Refl)}$$

$$\frac{\Gamma \vdash_S B_1 \leq A_1 : \star \qquad \Gamma \vdash_S A_2 \leq B_2 : \star}{\Gamma \vdash_S A_1 \to A_2 \leq_W B_1 \to B_2 : \star} \quad \text{(SWS-Arrow)}$$

$$\frac{\begin{array}{l}\Gamma, X{\leq}A_1{:}K \vdash_S A_2 \leq B_2 : \star \\ \Gamma \vdash_S A_1, B_1 \twoheadrightarrow_{\mathrm{n}} C : K'' \qquad \Gamma \vdash_S K, K' \twoheadrightarrow_{\mathrm{n}} K''\end{array}}{\Gamma \vdash_S \forall X{\leq}A_1{:}K.A_2 \leq_W \forall X{\leq}B_1{:}K'.B_2 : \star} \quad \text{(SWS-All)}$$

$$\frac{\begin{array}{ll} & \Gamma \vdash_S K_1, K_1' \twoheadrightarrow_{\mathrm{n}} K_1'' \\ \Gamma, X{\leq}A_1{:}K_1 \vdash_S A_2 \leq B_2 : K_2 & \Gamma \vdash_S A_1, B_1 \twoheadrightarrow_{\mathrm{n}} C : K_1''\end{array}}{\Gamma \vdash_S \Lambda X{\leq}A_1{:}K_1.A_2 \leq_W \Lambda X{\leq}B_1{:}K_1'.B_2 : \Pi X{\leq}C{:}K_1''.K_2} \quad \text{(SWS-TAbs)}$$

The weak head subtyping rules are motivated by the algorithmic rules in [17]. The rules SWS-Arrow, SWS-All, and SWS-TAbs are structural. The rule for the maximal type $\mathrm{T}_\star$ has a side condition to ensure that the algorithm is deterministic, and applications are only handled by SWS-TApp or SWS-Refl.

A particular instance of SWS-TApp is the rule for type variables. To check if $\Gamma \vdash_S X \leq_W A : K$, we have to check that the bound of X in Γ is a subtype of A, $\Gamma \vdash_S \Gamma(X) \leq A : K$, and since we know that A is in weak head normal form we save a recursive call of SS-Inc and normalize $\Gamma(X)$ in a premise. The side condition $A \not\equiv X$ is to ensure determinism; if $A \equiv X$, it follows by reflexivity.

3.5 Subtyping

$$\frac{\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} C : K \qquad \Gamma \vdash_S B \twoheadrightarrow_{\mathrm{w}} D : K \qquad \Gamma \vdash_S C \leq_W D : K}{\Gamma \vdash_S A \leq B : K} \quad \text{(SS-Inc)}$$

There is no rule for transitivity of subtyping in the semantic rules, but transitivity is a property of the “operational” subtyping (Lemma 4.21). Moreover, the rule SWS-TApp includes a step of transitivity along the bound of a variable in the context. We interleave weak head normalization steps in the subtyping algorithm via SS-Inc. An alternative formulation would weak head normalize the arguments of the hypothesis.

4 Metatheory for $\mathcal{F}^{\omega}_{\leq}$

In this section we prove fundamental properties about $\mathcal{F}^{\omega}_{\leq}$. We begin with results about the typed operational semantics and then show some basic results needed for the soundness proof for the system $\mathcal{F}^{\omega}_{\leq}$.

4.1 Metatheory for the Typed Operational Semantics

Here, the typed operational semantics is playing the same role as the algorithm in the usual development of the metatheory, but it also allows us to show results such as subject reduction and strong normalization for types.

Definition 4.1 (*Closed*)

1. A term M is closed with respect to a context Γ if $\mathrm{FV}(M) \cup \mathrm{FTV}(M) \subseteq \mathrm{dom}(\Gamma)$.
2. A type A is closed with respect to a context Γ if $\mathrm{FTV}(A) \subseteq \mathrm{dom}(\Gamma)$.

Judgements are closed if each of the terms to the right of the turnstile is closed with respect to the context.

Lemma 4.2 (*Closure*) If $\Gamma \vdash_S J$ then $\Gamma \vdash_S J$ is closed.

Lemma 4.3 (*Weak Head and Normal Forms*)

1. If $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$ then K' is in normal form.
2. If $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K$ then B is in weak head normal form and C and K are in normal form.

Proof: By simultaneous induction on derivations. □

Lemma 4.4 (*Context Formation*) If $\Gamma \vdash_S J$ then there is a (not necessarily strict) subderivation of $\Gamma \vdash_S \mathrm{ok}$.

Proof: By induction on derivations. □

Lemma 4.5 (*Determinacy*) If $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K$ and $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} D \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} E : K'$ then $B \equiv D$, $C \equiv E$ and $K \equiv K'$.

Proof: By induction on derivations. □

As we mentioned in the introduction, we want to prove completeness with respect to a system without the structural rules in Section 2.3. We shall write $\Gamma \vdash^{-} J$ for judgements in the restricted system without these rules.

Proposition 4.6 (*Completeness*)

1. $\Gamma \vdash_S \mathrm{ok}$ implies $\Gamma \vdash^{-} \mathrm{ok}$.
2. $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$ implies $\Gamma \vdash^{-} K$ and $\Gamma \vdash^{-} K =_{\beta} K'$.
3. $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K$ implies $\Gamma \vdash^{-} A : K$, $\Gamma \vdash^{-} A =_{\beta} B : K$, $\Gamma \vdash^{-} A =_{\beta} C : K$, and $\Gamma \vdash^{-} K =_{\beta} K$.
4. $\Gamma \vdash_S A \leq_W B : K$ implies $\Gamma \vdash^{-} A \leq B : K$.
5. $\Gamma \vdash_S A \leq B : K$ implies $\Gamma \vdash^{-} A, B : K$, $\Gamma \vdash^{-} A \leq B : K$ and $\Gamma \vdash^{-} K =_{\beta} K$.

PROOF: By simultaneous induction on derivations. We proceed by case analysis on the last rule of the derivation, presenting here a few representative cases. The hypotheses in each case are left implicit and follow exactly the notation of the rules in Section 3.

1. SC-EMPTY Immediate by C-EMPTY.

   SC-VAR By the induction hypothesis 3, $\Gamma \vdash A : \star$ and, by C-VAR, the result follows.

   SC-TVAR By the induction hypothesis 3,$\Gamma \vdash A : K'$, by the induction hypothesis 2, $\Gamma \vdash^- K =_\beta K'$. By the symmetry of kind equality and T-CONV, $\Gamma \vdash^- A : K$, and, by C-TVAR, the result follows.

2. SK-$\star$ By the induction hypothesis 1 $\Gamma \vdash^-$ ok. By K-$\star$, $\Gamma \vdash^- \star$, and, by K-EQ-$\star$, $\Gamma \vdash^- \star =_\beta \star$.

   SK-Π By the induction hypothesis 2, $\Gamma,\, X{\leq}A{:}K_1 \vdash^- K_2$, by the induction hypothesis 3, $\Gamma \vdash^- A : K_1'$. By the induction hypothesis 2, $\Gamma \vdash^- K_1 =_\beta K_1'$. By the symmetry of kind equality and T-CONV, $\Gamma \vdash^- A : K_1$. Then, by K-Π, $\Gamma \vdash^- \Pi X{\leq}A{:}K_1.K_2$.
   By the induction hypothesis 3, $\Gamma \vdash^- A =_\beta B : K_1$, and by the induction hypothesis 2 $\Gamma,\, X{\leq}A{:}K_1 \vdash^- K_2 =_\beta K_2'$. By K-EQ-$\Pi$, $\Gamma \vdash^- \Pi X{\leq}A{:}K_1.K_2 =_\beta \Pi X{\leq}B{:}K_1'.K_2'$.

3. ST-TAPP By the induction hypothesis 3, $\Gamma \vdash^- A : \Pi X{\leq}C{:}K_1.K_2$ and $\Gamma \vdash^- B =_\beta E : K_1$. By the induction hypothesis 5, $\Gamma \vdash^- E \leq C : K_1$. By S-CONV, $\Gamma \vdash^- B \leq E : K_1$, and, by S-TRANS, $\Gamma \vdash^- B \leq C : K_1$. By T-TAPP, $\Gamma \vdash^- A\,B : K_2[X{\leftarrow}B]$. By the induction hypothesis 2, $\Gamma \vdash^- K_2[X{\leftarrow}B] =_\beta K$, and, by T-CONV, $\Gamma \vdash^- A\,B : K$. We now have to prove that $\Gamma \vdash^- A\,B =_\beta D\,F : K$. By the induction hypothesis 3, $\Gamma \vdash^- A =_\beta D : \Pi X{\leq}C{:}K_1.K_2$. By T-EQ-TAPP, $\Gamma \vdash^- A\,B =_\beta D\,F : K_2[X{\leftarrow}B]$, and by T-EQ-CONV, $\Gamma \vdash^- A\,B =_\beta D\,F : K$.

4. SWS-ALL By inductive hypothesis 5, $\Gamma,\, X{\leq}A_1{:}K \vdash^- A_2 \leq B_2 : \star$ and $\Gamma,\, X{\leq}A_1{:}K \vdash^- B_2 : \star$. Then, by S-ALL, $\Gamma \vdash^- \forall X{\leq}A_1{:}K.A_2 \leq \forall X{\leq}A_1{:}K.B_2 : \star$. We now want to prove using T-EQ-ALL that $\Gamma \vdash^- \forall X{\leq}A_1{:}K.B_2 \leq \forall X{\leq}B_1{:}K'.B_2 : \star$. The result then follows by S-CONV and S-TRANS. For that we need:

   (a) $\Gamma,\, X{\leq}A_1{:}K \vdash^- B_2 =_\beta B_2 : \star$, which follows from T-EQ-REFL.

   (b) $\Gamma \vdash^- K =_\beta K'$, which follows from the induction hypothesis 2, K-EQ-SYM and K-EQ-TRANS.

   (c) $\Gamma \vdash^- A_1 =_\beta B_1 : K$. By the induction hypothesis 2 and K-EQ-SYM, $\Gamma \vdash^- K'' =_\beta K$. By the induction hypothesis 3, $\Gamma \vdash^- A_1 =_\beta B_1 : K''$, and by T-EQ-CONV, $\Gamma \vdash^- A_1 =_\beta B_1 : K$.

5. SS-INC By the induction hypothesis 3, $\Gamma \vdash A, B, C, D : K$, $\Gamma \vdash^- A =_\beta C : K$, $\Gamma \vdash^- B =_\beta D : K$, and $\Gamma \vdash^- K =_\beta K$. By S-CONV, $\Gamma \vdash^- A \leq C : K$ and $\Gamma \vdash^- D \leq B : K$, and, by the induction hypothesis 4, $\Gamma \vdash^- C \leq D : K$. Finally, by S-TRANS, $\Gamma \vdash^- A \leq B : K$. □

In Section 5 we shall see how Soundness (Corollary 5.12) can be used together with this result to show the admissibility of the structural rules.

The following technique for proving Thinning comes from McKinna and Pollack's development of the metatheory of Pure Type Systems [29].

DEFINITION 4.7 (*Parallel Type Substitution*) A *parallel type substitution* γ for Γ is an assignment of types to type variables in $\mathrm{dom}(\Gamma)$. We write $\gamma[X{:=}A]$ to extend γ to assign A to X, $\gamma(X)$ for the value of the assignment at a variable, and $A[\gamma]$ for the replacement of variables in A with the values in γ; the value of this is undefined if there is a variable in A not in $\mathrm{dom}(\gamma)$. We say that γ is a type substitution for Γ in Δ if $\Delta \vdash_S \gamma(X) \leq A[\gamma] : K'$, where $\Delta \vdash_S K[\gamma] \twoheadrightarrow_{\mathrm{n}} K'$, for each X with $\Gamma = \Gamma_1, X{\leq}A : K, \Gamma_2$. A *renaming* is a parallel type substitution of variables for variables.

LEMMA 4.8 (*Renaming*) If $\Gamma \vdash_S J$ and δ is a renaming for Γ in Δ then $\Delta \vdash_S J[\delta]$.

PROOF: By induction on derivations. □

We write $\vdash_S \Delta \geq \Gamma$ if $\Delta \vdash_S \mathrm{ok}$, $x{:}A \in \Gamma$ implies $x{:}A \in \Delta$, and $X{\leq}A{:}K \in \Gamma$ implies $X{\leq}A{:}K \in \Delta$. Thinning, which says that judgements are monotonic with respect to context extension, now follows as a corollary of Renaming taking δ to be the identity substitution.

COROLLARY 4.9 (*Thinning*) If $\Gamma \vdash_S J$ and $\vdash_S \Delta \geq \Gamma$ then $\Delta \vdash_S J$.

LEMMA 4.10 (*Adequacy*)

- If $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$ then $K \twoheadrightarrow_{\beta_2} K'$.
- If $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K$ then $A \twoheadrightarrow_{\beta_2} B \twoheadrightarrow_{\beta_2} C$.

We use parallel reduction [28, 29] as a tool for proving subject reduction for the typed operational semantics.

DEFINITION 4.11 (*Parallel Reduction*) Parallel reduction $\Rightarrow$ is the least relation over types and kinds defined by the following rules of inference.

$$X \Rightarrow X \qquad \text{(P-VAR)}$$

$$\frac{A \Rightarrow A' \qquad K \Rightarrow K' \qquad B \Rightarrow B'}{\Lambda X{\leq}A{:}K.B \Rightarrow \Lambda X{\leq}A'{:}K'.B'} \qquad \text{(P-LAMBDA)}$$

$$\frac{A \Rightarrow A' \qquad B \Rightarrow B'}{A\,B \Rightarrow A'\,B'} \qquad \text{(P-APP)}$$

$$\frac{A \Rightarrow A' \qquad B \Rightarrow B'}{(\Lambda X{\leq}C{:}K.A)(B) \Rightarrow A'[X{\leftarrow}B']} \qquad \text{(P-BETA)}$$

plus similar rules, allowing reduction on each of the subterms, for the other type and kind formers.

Parallel reduction extends in the obvious way to contexts.

Parallel reduction is useful because it is closed under the following rule of substitution:

$$\frac{A \Rightarrow A' \qquad B \Rightarrow B'}{A[X\leftarrow B] \Rightarrow A'[X\leftarrow B']} \tag{P-Subst}$$

The following proof uses this and other simple properties about parallel reduction. See Takahashi's excellent account of parallel reduction [35] for more details.

Lemma 4.12 (*Parallel Subject Reduction for Types and Kinds*)

- If $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \ {}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K$, $\Gamma \Rightarrow \Gamma'$ and $A \Rightarrow A'$, then there is a B' such that $B \Rightarrow B'$ and $\Gamma' \vdash_S A' \twoheadrightarrow_{\mathrm{w}} B' \ {}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} C : K$.
- If $\Gamma \vdash_S A \leq_W B : K$, $\Gamma \Rightarrow \Gamma'$, $A \Rightarrow A'$, and $B \Rightarrow B'$ then $\Gamma' \vdash_S A' \leq_W B' : K$.
- If $\Gamma \vdash_S A \leq B : K$, $\Gamma \Rightarrow \Gamma'$, $A \Rightarrow A'$, and $B \Rightarrow B'$ then $\Gamma' \vdash_S A' \leq B' : K$.

Proof: By induction on derivations. We consider several cases:

ST-TVar Suppose $\Gamma \Rightarrow \Gamma'$ and $X \Rightarrow B$. By inversion $B \equiv X$. Also, clearly if $(X \leq A : K) \in \Gamma$ then $\Gamma \Rightarrow \Gamma'$ implies $A \Rightarrow A'$, $K \Rightarrow K''$ and $(X \leq A' : K') \in \Gamma'$. Hence, by inductive hypothesis $\Gamma' \vdash_S A' : K'$ and $\Gamma' \vdash_S K'' \twoheadrightarrow_{\mathrm{n}} K'$, so $\Gamma' \vdash_S X \twoheadrightarrow_{\mathrm{w}} X \twoheadrightarrow_{\mathrm{n}} X : K'$.

ST-TApp Suppose $\Gamma \Rightarrow \Gamma'$ and $A\,B \Rightarrow E$. Clearly if D is not an abstraction then A is not an abstraction, so by inversion $E \equiv A'\,B'$ with $A \Rightarrow A'$ and $B \Rightarrow B'$. Hence, by inductive hypothesis there is a D' such that $\Gamma' \vdash_S A' \twoheadrightarrow_{\mathrm{w}} D' \twoheadrightarrow_{\mathrm{n}} D : \Pi X{\leq}C{:}K_1.K_2$ and $D \Rightarrow D'$, and because D is not an abstraction and D is weak head normal then D' is not an abstraction; there is an E' such that $\Gamma' \vdash_S B' \twoheadrightarrow_{\mathrm{w}} E' \twoheadrightarrow_{\mathrm{n}} F : K_1$ and $E \Rightarrow E'$; $\Gamma' \vdash_S K_2[X\leftarrow B'] \twoheadrightarrow_{\mathrm{n}} K$; and $\Gamma' \vdash_S E' \leq_W C : K_1$. Hence, $\Gamma' \vdash_S A'\,B' \twoheadrightarrow_{\mathrm{w}} D'\,F \twoheadrightarrow_{\mathrm{n}} D'\,F$ by ST-TApp.

ST-Beta Suppose $\Gamma \Rightarrow \Gamma'$ and $B\,C \Rightarrow G$. By inversion of the reduction:

- $B \Rightarrow B'$, $C \Rightarrow C'$, and $G \equiv B'\,C'$. Then by inductive hypothesis $\Gamma' \vdash_S B' \twoheadrightarrow_{\mathrm{w}} \Lambda X{\leq}A''{:}K_1''.D' : \Pi X{\leq}A'{:}K_1'.K_2$ with $\Lambda X{\leq}A{:}K_1.D \Rightarrow \Lambda X{\leq}A''{:}K_1''.D'$, and also $\Gamma' \vdash_S C' \leq A'' : K_1'$, $\Gamma' \vdash_S K_2[X\leftarrow C'] \twoheadrightarrow_{\mathrm{n}} K$, and there is an E' such that $\Gamma' \vdash_S D'[X\leftarrow C'] \twoheadrightarrow_{\mathrm{w}} E' \twoheadrightarrow_{\mathrm{n}} F : K$ and $E \Rightarrow E'$. Hence $\Gamma' \vdash_S B'\,C' \twoheadrightarrow_{\mathrm{w}} E' \twoheadrightarrow_{\mathrm{n}} F : K$ with $E \Rightarrow E'$.
- $B \equiv \Lambda X{\leq}H{:}K_3.I$, $I \Rightarrow I'$, $C \Rightarrow C'$ and $B\,C \Rightarrow I'[X\leftarrow C']$. By inversion of the premise for B we know that $H \equiv A$, $K_3 \equiv K_1$ and $I \equiv D$. Hence $D \Rightarrow D'$ and $C \Rightarrow C'$ imply $D[X\leftarrow C] \Rightarrow D'[X\leftarrow C']$, so by inductive hypothesis there is an E' such that $\Gamma' \vdash_S D'[X\leftarrow C'] \twoheadrightarrow_{\mathrm{w}} E' \twoheadrightarrow_{\mathrm{n}} F : K$ and $E \Rightarrow E'$.

SWS-Refl By Lemma 4.3 and the definition of weak head normal, $X(A_1, \ldots, A_m)$ is normal. Hence, $X(A_1, \ldots, A_m) \Rightarrow C$ implies $C \equiv X(A_1, \ldots, A_m)$, and by inductive hypothesis $\Gamma \Rightarrow \Gamma'$ implies $\Gamma' \vdash_S X(A_1, \ldots, A_m) \ {}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} B : K$. Hence, $\Gamma' \vdash_S C \leq_W D : K$ if $X(A_1, \ldots, A_m) \Rightarrow C$ and $X(A_1, \ldots, A_m) \Rightarrow D$. □

It is easy to show that $\rightarrow_\beta$ is included in $\Rightarrow$, and that $\Rightarrow$ is included in $\twoheadrightarrow_\beta$, so we have the following corollary:

COROLLARY 4.13 (*Subject Reduction for Types and Kinds*)

- If $\Gamma \vdash_S \mathrm{ok}$ and $\Gamma \rightarrow_\beta \Gamma'$ then $\Gamma' \vdash_S \mathrm{ok}$.
- If $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}{\twoheadrightarrow_{\mathrm{n}}}\; C : K$ and $A \rightarrow_\beta A'$ then there is a B' such that $B \twoheadrightarrow_\beta B'$ and $\Gamma \vdash_S A' \twoheadrightarrow_{\mathrm{w}} B' \;{}_{\mathrm{w}}{\twoheadrightarrow_{\mathrm{n}}}\; C : K$.
- If $\Gamma \vdash_S A \leq B : K$, $A \rightarrow_\beta A'$ and $B \rightarrow_\beta B'$ then $\Gamma \vdash_S A' \leq B' : K$.

Notice that this corollary incorporates both Subject Reduction and Church–Rosser, because the normal form is preserved by any one-step reduction.

We now prove Strong Normalization, using Subject Reduction to help.

DEFINITION 4.14 (*Strong Normalization*) Strong normalization for types, written $\mathrm{SN}(A)$, is the least relation closed under the following rule of inference:

$$\frac{\text{for all } B.(A \rightarrow_{\beta_2} B) \implies \mathrm{SN}(B)}{\mathrm{SN}(A)} \tag{SN-I}$$

and similarly for kinds.

Strong normalization is easily seen to be closed under $\rightarrow_{\beta_2}$-reduction.

LEMMA 4.15 (*Strong Normalization for Types and Kinds*)

1. If $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}{\twoheadrightarrow_{\mathrm{n}}}\; C : K$ then A is strongly normalizing.
2. If $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$ then K is strongly normalizing.

PROOF: By induction on derivations.

ST-$\star$ By SN-I we need to show that if $\star \rightarrow_{\beta_2} K'$ then $\mathrm{SN}(K')$, which follows because $\star \rightarrow_{\beta_2} K'$ is impossible.

ST-Π By inductive hypothesis we know that $\mathrm{SN}(A)$, $\mathrm{SN}(K_1)$ and $\mathrm{SN}(K_2)$. By induction on these premises we show that $\mathrm{SN}(\Pi X{\leq}A{:}K_1.K_2)$. By SN-I we need to show that if $\Pi X{\leq}A{:}K_1.K_2 \rightarrow_{\beta_2} K'$ then $\mathrm{SN}(K')$. There are three possible reductions, corresponding to the subterms of $\Pi X{\leq}A{:}K_1.K_2$, and each of these cases follows by the appropriate inductive hypothesis.

ST-TOP By SN-I and the impossibility of $\mathrm{T}_\star \rightarrow_{\beta_2} B$.

ST-TVAR By SN-I and the impossibility of $X \rightarrow_{\beta_2} B$.

ST-TApp By inductive hypothesis we know that SN(A) and SN(B). By induction on these we show that if $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} D : \Pi X{\leq}C{:}K_1.K_2$ and D is not an abstraction then SN($A\,B$). By SN-I we need to show that if $A\,B \rightarrow_{\beta_2} G$ then SN(G). Again, if D is not an abstraction then A is not an abstraction, so if $A\,B \rightarrow_{\beta_2} G$ then we have two cases:

- $A \rightarrow_{\beta_2} A'$. Then by Subject Reduction (Lemma 4.13) there is a D' such that $\Gamma \vdash_S A' \twoheadrightarrow_{\mathrm{w}} D' : \Pi X{\leq}C{:}K_1.K_2$ and $D \Rightarrow D'$. Clearly if $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} D : \Pi X{\leq}C{:}K_1.K_2$, D is not an abstraction and $D \Rightarrow D'$ then D' is not an abstraction. Hence, by inductive hypothesis we know SN($A'\,B$).
- $B \rightarrow_{\beta_2} B'$. This follows directly by inductive hypothesis.

ST-Arrow By inductive hypothesis we know SN(A_1) and SN(A_2). By induction on these we show SN($A_1{\rightarrow}A_2$). By SN-I we need to show that if $A_1{\rightarrow}A_2 \rightarrow_{\beta_2} C$ then SN(C). By inversion either $A_1 \rightarrow_{\beta_2} C_1$ or $A_2 \rightarrow_{\beta_2} C_2$, and each case follows by the inductive hypothesis.

ST-TAll Similar to the case for ST-Π.

ST-TAbs Similar to the case for ST-Π.

ST-Beta By inductive hypothesis we know that SN(B), SN(C), and SN($D[X{\leftarrow}C]$). By induction on SN(B) and SN(C) we show that $\Gamma \vdash_S B \twoheadrightarrow_{\mathrm{w}} \Lambda X{\leq}A{:}K_1.D : \Pi X{\leq}A'{:}K_1'.K_2$ and SN($D[X{\leftarrow}C]$) imply SN($B\,C$). By SN-I, we need $B\,C \rightarrow_{\beta_2} G$ implies SN(G). By inversion of $B\,C \rightarrow_{\beta_2} G$ there are three cases:

- $B \rightarrow_{\beta_2} B'$. Then by Subject Reduction (Lemma 4.13) there is an H such that $\Gamma \vdash_S B' \twoheadrightarrow_{\mathrm{w}} H : \Pi X{\leq}A'{:}K_1'.K_2$ and $\Lambda X{\leq}A{:}K_1.D \Rightarrow H$. By inversion $H \equiv \Lambda X{\leq}A''{:}K_1''.D'$ with $A \Rightarrow A''$, $K_1 \Rightarrow K_1''$ and $D \Rightarrow D'$. Because SN is closed under reduction we know SN($D'[X{\leftarrow}C]$). Hence SN($B'\,C$) by inductive hypothesis.
- $C \rightarrow_{\beta_2} C'$. Then since SN is closed under reduction SN($D[X{\leftarrow}C']$), so by inductive hypothesis SN($B\,C'$).
- $B \equiv \Lambda X{\leq}A''{:}K_1''.H$ and $B\,C \rightarrow_{\beta_2} H[X{\leftarrow}C]$. By inversion of $\Gamma \vdash_S B \twoheadrightarrow_{\mathrm{w}} \Lambda X{\leq}A{:}K_1.D : \Pi X{\leq}A'{:}K_1'.K_2$ we know that $B \equiv \Lambda X{\leq}A{:}K_1.D$, so in particular $H \equiv D$. Hence SN($H[X{\leftarrow}C]$) follows by assumption. □

Definition 4.16 We define conversion of contexts, written $\vdash_S \Gamma;\Delta \twoheadrightarrow_{\mathrm{n}} \Phi$, as the least relation closed under the following rules of inference:

$$\vdash_S \emptyset;\emptyset \twoheadrightarrow_{\mathrm{n}} \emptyset \qquad \text{(SCN-Empty)}$$

$$\frac{\vdash_S \Gamma;\Delta \twoheadrightarrow_{\mathrm{n}} \Phi \qquad \Gamma \vdash_S A, B \twoheadrightarrow_{\mathrm{n}} C : \star \qquad x \notin \mathrm{dom}(\Gamma)}{\vdash_S \Gamma,\, x{:}A;\Delta,\, x{:}B \twoheadrightarrow_{\mathrm{n}} \Phi,\, x{:}C} \qquad \text{(SCN-Var)}$$

$$\frac{\vdash_S \Gamma;\Delta \twoheadrightarrow_{\mathrm{n}} \Phi \qquad \Gamma \vdash_S K, K' \twoheadrightarrow_{\mathrm{n}} K'' \qquad \Gamma \vdash_S A, B \twoheadrightarrow_{\mathrm{n}} C : K'' \quad X \notin \mathrm{dom}(\Gamma)}{\vdash_S \Gamma,\, X{\leq}A{:}K;\Delta,\, X{\leq}B{:}K' \twoheadrightarrow_{\mathrm{n}} \Phi,\, X{\leq}C{:}K''} \tag{SCN-TVar}$$

Lemma 4.17 (*Context Conversion*) If $\vdash_S \Gamma, \Delta \twoheadrightarrow_{\mathrm{n}} \Phi$ and $\Gamma \vdash_S J$ then $\Delta \vdash_S J$.

Proof: By induction on derivations. We consider two representative cases:

ST-TVar We know that $(X{\leq}A : K) \in \Gamma$, so by inversion of $\Gamma;\Delta \twoheadrightarrow_{\mathrm{n}} \Phi$ we know that there are Γ_0, Γ_1, Δ_0 and Δ_1 such that $\Gamma \equiv \Gamma_0, X{\leq}A : K, \Gamma_1$ and $\Delta \equiv \Delta_0, X{\leq}B : K'', \Delta_1$, where $\Gamma_0 \vdash_S A \twoheadrightarrow_{\mathrm{n}} C : K'''$, $\Gamma_0 \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'''$, $\Delta_0 \vdash_S B \twoheadrightarrow_{\mathrm{n}} C : K'''$, and $\Delta_0 \vdash_S K'' \twoheadrightarrow_{\mathrm{n}} K'''$. By Renaming $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'''$, $\Delta \vdash_S K'' \twoheadrightarrow_{\mathrm{n}} K'''$, and $\Delta \vdash_S B \twoheadrightarrow_{\mathrm{n}} C : K'''$. We have a premise that $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$, so by Determinacy $K' \equiv K'''$, so $\Delta \vdash_S X \twoheadrightarrow_{\mathrm{w}} X \twoheadrightarrow_{\mathrm{n}} X : K'$.

ST-TAbs By inductive hypothesis $\Delta \vdash_S A \twoheadrightarrow_{\mathrm{n}} C : K_1'$ and $\Delta \vdash_S K_1 \twoheadrightarrow_{\mathrm{n}} K_1'$. Hence $\vdash_S \Gamma,\, X{\leq}A : K_1; \Delta,\, X{\leq}B : K_1 \twoheadrightarrow_{\mathrm{n}} \Phi,\, X{\leq}C : K_1'$, and so by inductive hypothesis $\Delta,\, X{\leq}A : K_1 \vdash_S B \twoheadrightarrow_{\mathrm{n}} D : K_2$. The result follows by ST-TAbs. □

Lemma 4.18 (*Subtyping Conversion*)

- Suppose that $\Gamma \vdash_S A \leq_W B : K$. Then:
  - If $\Gamma \vdash_S A, A' \ {}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}} C : K$ then $\Gamma \vdash_S A' \leq_W B : K$.
  - If $\Gamma \vdash_S B, B' \ {}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}} C : K$ then $\Gamma \vdash_S A \leq_W B' : K$.
- Suppose that $\Gamma \vdash_S A \leq B : K$. Then:
  - If $\Gamma \vdash_S A, A' \twoheadrightarrow_{\mathrm{n}} C : K$ then $\Gamma \vdash_S A' \leq B : K$.
  - If $\Gamma \vdash_S B, B' \twoheadrightarrow_{\mathrm{n}} C : K$ then $\Gamma \vdash_S A \leq B' : K$.

Proof: By induction on derivations. We show two interesting cases:

SWS-All We consider the case that $\Gamma \vdash_S \forall X{\leq}A_1{:}K.A_2, A' \ {}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}} D : \star$, where the other case is similar but simpler. By inversion on the derivation for $\forall X{\leq}A_1{:}K.A_2$ we know that $D \equiv \forall X{\leq}C_1{:}K'''.C_2$ with $\Gamma \vdash_S A_1 \twoheadrightarrow_{\mathrm{n}} C_1 : K'''$, $\Gamma \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'''$ and $\Gamma,\, X{\leq}A_1 : K \vdash_S A_2 \twoheadrightarrow_{\mathrm{n}} C_2 : \star$. By Determinacy $K'' \equiv K'''$ and $C \equiv C_1$. By inversion on the derivation that $\Gamma \vdash_S A' \ {}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}} \forall X{\leq}C_1{:}K'''.C_2 : \star$ we know that $A' \equiv \forall X{\leq}A_1'{:}K''''.A_2'$, $\Gamma \vdash_S A_1' \twoheadrightarrow_{\mathrm{n}} C_1 : K''$, $\Gamma \vdash_S K'''' \twoheadrightarrow_{\mathrm{n}} K''$ and $\Gamma,\, X{\leq}A_1' : K'''' \vdash_S A_2' \twoheadrightarrow_{\mathrm{n}} C_2 : \star$. By Context Conversion we know that $\Gamma,\, X{\leq}A_1 : K \vdash_S A_2' \twoheadrightarrow_{\mathrm{n}} C_2 : \star$, so by inductive hypothesis $\Gamma,\, X{\leq}A_1 : K \vdash_S A_2' \leq B_2 : \star$. Finally, by Context Conversion again $\Gamma,\, X{\leq}A_1' : K'''' \vdash_S A_2' \leq B_2 : \star$, and the result follows by SWS-All.

SS-Inc We consider the case that $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} C' \;{}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}}\; E : K$ and $\Gamma \vdash_S A' \twoheadrightarrow_{\mathrm{w}} C'' \;{}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}}\; E : K$, where the other case is similar. We know $C \equiv C'$ by Determinacy. Furthermore, by Adequacy and Subject Reduction $\Gamma \vdash_S C, C'' \;{}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}}\; E : K$, so by inductive hypothesis $\Gamma \vdash_S C'' \leq_W D : K$. Hence $\Gamma \vdash_S A' \leq B : K$ by SS-Inc. □

Lemma 4.19 If $\Gamma \vdash_S A \leq_W B : K$ then there is a C such that $\Gamma \vdash_S A \;{}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}}\; C : K$.

Proof: By inversion of $\Gamma \vdash_S A \leq_W B : K$, also using inversion on the premises for SWS-Arrow, SWS-All and SWS-TAbs. □

Lemma 4.20 (*Reflexivity*) If $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{n}} B : K$ then $\Gamma \vdash_S A \leq A : K$.

Proof: We show the stronger property, that if $\Gamma \vdash_S A \twoheadrightarrow_{\mathrm{w}} B \;{}_{\mathrm{w}}{\twoheadrightarrow}_{\mathrm{n}}\; C : K$ then $\Gamma \vdash_S B \leq_W B : K$ and $\Gamma \vdash_S A \leq A : K$, by induction on derivations. □

Lemma 4.21 (*Transitivity*) If $\Gamma \vdash_S A \leq B : K$ and $\Gamma \vdash_S B \leq C : K$ then $\Gamma \vdash_S A \leq C : K$.

Proof: We show the stronger property that:

- if $\Gamma \vdash_S B \leq C : K$ then:
  1. if $\Gamma \vdash_S A \leq B : K$ then $\Gamma \vdash_S A \leq C : K$, and
  2. if $\Gamma \vdash_S C \leq D : K$ then $\Gamma \vdash_S B \leq D : K$, and
- if $\Gamma \vdash_S B \leq_W C : K$ then:
  1. if $\Gamma \vdash_S A \leq_W B : K$ then $\Gamma \vdash_S A \leq_W C : K$, and
  2. if $\Gamma \vdash_S C \leq_W D : K$ then $\Gamma \vdash_S B \leq_W D : K$.

by induction on derivations. We show several cases:

SWS-Top Case 1 follows by Lemma 4.19 and SWS-Top, and Case 2 follows by inversion on derivations such that $\Gamma \vdash_S \mathrm{T}_\star \leq_W D : \star$.

SWS-Arrow

1. Suppose $\Gamma \vdash_S C \leq_W A_1 \to A_2 : \star$. By induction on this we show that $\Gamma \vdash_S C \leq_W B_1 \to B_2 : \star$.

   SWS-TApp By the second inductive hypothesis and SWS-TApp.

   SWS-Arrow We have that $C \equiv C_1 \to C_2$ and that $\Gamma \vdash_S C \leq_W A_1 \to A_2 : \star$ by $\Gamma \vdash_S A_1 \leq C_1 : \star$ and $\Gamma \vdash_S C_2 \leq A_2 : \star$. By the first inductive hypothesis Case 2 $\Gamma \vdash_S B_1 \leq C_1 : \star$, and by the first inductive hypothesis Case 1 $\Gamma \vdash_S C_2 \leq B_2 : \star$. Hence $\Gamma \vdash_S C_1 \to C_2 \leq_W B_1 \to B_2 : \star$ by SWS-Arrow.

2. Suppose $\Gamma \vdash_S B_1 \to B_2 \leq_W C : \star$. By inversion on this we show that $\Gamma \vdash_S A_1 \to A_2 \leq_W C : \star$.

   SWS-TOP Then $\Gamma \vdash_S A_1 \to A_2 \;{}_{\mathrm{w}}\!\twoheadrightarrow_{\mathrm{n}}\; D : \star$ by Lemma 4.19, so $\Gamma \vdash_S A_1 \to A_2 \leq_W \mathrm{T}_\star : \star$ by SWS-TOP.

   SWS-ARROW We know that $\Gamma \vdash_S C_1 \leq B_1 : \star$ and $\Gamma \vdash_S B_2 \leq C_2 : \star$. Hence by inductive hypothesis Case 1 $\Gamma \vdash_S C_1 \leq A_1 : \star$, and by the first inductive hypothesis Case 2 $\Gamma \vdash_S A_2 \leq C_2 : \star$, and so $\Gamma \vdash_S A_1 \to A_2 \leq_W C_1 \to C_2 : \star$ by SWS-ARROW.

SS-INC Both cases follow by inversion of the assumption, Determinacy, the appropriate inductive hypothesis for $\leq_W$ and SS-INC. □

4.2 Basic Properties For $\mathcal{F}^{\omega}_{\leq}$

LEMMA 4.22 If $\Gamma \vdash \mathrm{ok}$ then all variables in $\mathrm{dom}(\Gamma)$ are different.

PROOF: By induction on the structure of Γ. □

LEMMA 4.23 If $\Gamma \vdash J$ and Γ' is a prefix of Γ, then $\Gamma' \vdash \mathrm{ok}$.

PROOF: By induction on the derivation of $\Gamma \vdash J$. □

LEMMA 4.24 (*Strengthening*) If $\Gamma_1, y{:}C, \Gamma_2 \vdash J$ and $y \notin \mathrm{FV}(J)$ then $\Gamma_1, \Gamma_2 \vdash J$.

PROOF: By induction on the derivation of $\Gamma_1, y{:}C, \Gamma_2 \vdash J$. Most cases follow by the induction hypothesis and the corresponding rule, and C-VAR uses Lemma 4.23. □

LEMMA 4.25 If $Y \notin \mathrm{FV}(A)$ then $B[X{\leftarrow}C][Y{\leftarrow}A[X{\leftarrow}C]] = B[Y{\leftarrow}A][X{\leftarrow}C]$.

PROOF: By induction on the structure of B. □

LEMMA 4.26 (*Term Substitution*)

1. If $\Gamma_1, x{:}A, \Gamma_2 \vdash M : B$ and $\Gamma_1 \vdash N : A$ then $\Gamma_1, \Gamma_2 \vdash M[x{\leftarrow}N] : B$.
2. If $\Gamma_1, X{\leq}A{:}K, \Gamma_2 \vdash M : B$ and $\Gamma_1 \vdash C \leq A$ then $\Gamma_1, \Gamma_2[X{\leftarrow}C] \vdash M[X{\leftarrow}C] : B[X{\leftarrow}C]$.

PROOF:

1. By induction on derivations, where for T-VAR we use Weakening (Corollary 4.9) and Strengthening (Lemma 4.24), and for T-ABS we use Lemmas 4.23 and 4.22.
2. By induction on derivations, using Lemmas 4.23 and 4.22 in the case T-TABS; using Lemma 4.25 and the rule SUBST in the case T-TAPP; and using the rule SUBST in the case T-SUB. □

5 Soundness

In this section we show the most important result for the metatheory of $\mathcal{F}^{\omega}_{\leq}$: that the typed operational semantics is sound for the typing rules in Section 2. As we discussed in Section 1.4, this proof is essentially similar to traditional proofs of strong normalization, although it includes several technical modifications allowing us to prove soundness instead of normalization.

5.1 The Interpretation

We begin by defining the interpretation of kinds K with respect to a type substitution γ in a context Δ. There are two components to the interpretation: the first component is a set of types well-formed in Δ with particular properties, and models the judgement $\Gamma \vdash A : K$; the second component is a relation on types in the first component, and models the judgement $\Gamma \vdash A \leq B : K$.

Partial interpretations are common in defining the semantics of dependent type theories [24, 34]. In our proof, we need a partial interpretation to guarantee that the bound A is well-formed for each Π-constructor $\Pi X{\leq}A{:}K_1.K_2$. This is information that can only be known when the proof itself is carried out, not when we define the interpretation. We prove that the interpretation of a kind K is always defined if K is well-formed according to the typing rules of $\mathcal{F}^{\omega}_{\leq}$.

We need to include type information in our model, because we are proving soundness with respect to a system with types. Unfortunately, the approach used for simpler type systems, to assume an infinite collection of variables of each type [23], does not easily transfer to our system. The problem is that the kinds cannot be enumerated separately from the variables, because the kind $\Pi X{\leq}A{:}K_1.K_2$ may include occurrences of variables in the type A. Hence, we build a Kripke-style model following Coquand and Gallier [20], where the possible worlds are valid contexts Δ and ordering is lexicographic, written $\Delta' \geq \Delta$.

The interpretation satisfies conditions similar to the usual saturated set conditions and properties lifted from the typed operational semantics, such as transitivity elimination (Lemma 5.6); properties about Kripke-style models such as monotonicity (Lemma 5.7); and the substitution property (Lemma 5.8).

Definition 5.1 (*Semantic Object*) A is a *semantic object* for Γ and K if $\Gamma \vdash_S A \leq \mathrm{T}_K : K$.

Definition 5.2 (*Interpretation of Kinds*) We give a partial interpretation of kinds for both judgements, $[\![-]\!]_{:}(-,-)$ for typing and $[\![-]\!]_{\leq}(-,-)$ for subtyping:

- The interpretation $[\![\star]\!]\gamma\Delta$ is well-defined if $\Delta \vdash_S \mathrm{ok}$. The two components are:
  - $[\![\star]\!]_{:}\gamma\Delta = \{A \mid A \text{ is a semantic object for } \Delta \text{ and } \star\}$.
  - $[\![\star]\!]_{\leq}\gamma\Delta = \{(A, B) \mid A \text{ and } B \text{ are semantic objects and } \Delta \vdash_S A \leq B : \star\}$.

- The interpretation $[\![\Pi X{\leq}A{:}K_1.K_2]\!]\gamma\Delta$ is well-defined if the following conditions hold:
  - there is a K' such that $\Delta \vdash_S (\Pi X{\leq}A{:}K_1.K_2)[\gamma] \twoheadrightarrow_{\mathrm{n}} K'$,
  - $[\![K_1]\!]\gamma\Delta'$ is defined for any $\vdash_S \Delta' \geq \Delta$, and
  - if $\vdash_S \Delta' \geq \Delta$ and $(C, A[\gamma]) \in [\![K_1]\!]_{\leq}\gamma\Delta'$, then $[\![K_2]\!]\gamma[X{:=}C]\Delta'$ is well-defined.

  Under these circumstances, the two components are:
  - $[\![\Pi X{\leq}A{:}K_1.K_2]\!]_{:}\gamma\Delta$ is the set of B such that:
    - B is a semantic object for Δ and K',
    - if $\vdash_S \Delta' \geq \Delta$ and $(C, A[\gamma]) \in [\![K_1]\!]_{\leq}\gamma\Delta'$, then $B\,C \in [\![K_2]\!]_{:}\gamma[X{:=}C]\Delta'$.
  - $[\![\Pi X{\leq}A{:}K_1.K_2]\!]_{\leq}\gamma\Delta$ is the set of (B, C) such that:
    - B and C are in $[\![\Pi X{\leq}A{:}K_1.K_2]\!]_{:}\gamma\Delta$,
    - $\Delta \vdash_S B \leq C : K'$,
    - if $\vdash_S \Delta' \geq \Delta$ and $(D, A[\gamma]) \in [\![K_1]\!]_{\leq}\gamma\Delta'$ then $(B\,D, C\,D) \in [\![K_2]\!]_{\leq}\gamma[X{:=}D]\Delta'$.

DEFINITION 5.3 (*Interpretation of Contexts*) We define a partial interpretation of contexts:

- $[\![\emptyset]\!]\Delta = \{\epsilon\}$. This is defined if $\Delta \vdash_S$ ok.
- $[\![\Gamma, x{:}A]\!]\Delta = [\![\Gamma]\!]\Delta$. This is defined if $x \notin \mathrm{dom}(\Gamma)$ and, for any $\vdash_S \Delta' \geq \Delta$, $[\![\Gamma]\!]\Delta'$ is defined and $A[\gamma] \in [\![\star]\!]_{:}\gamma\Delta'$ for every $\gamma \in [\![\Gamma]\!]\Delta'$.
- $[\![\Gamma, X{\leq}A{:}K]\!]\Delta = \{\gamma[X{:=}B] \mid \gamma \in [\![\Gamma]\!]\Delta \text{ and } (B, A[\gamma]) \in [\![K]\!]_{\leq}\gamma\Delta\}$. This is defined if $X \notin \mathrm{dom}(\Gamma)$ and, for any $\vdash_S \Delta' \geq \Delta$, $[\![\Gamma]\!]\Delta'$ is defined and $A[\gamma] \in [\![K]\!]_{:}\gamma\Delta'$ for every $\gamma \in [\![\Gamma]\!]\Delta'$.

5.2 Properties of the Interpretation

We need to establish a variety of properties about the interpretation before carrying out the soundness proof.

DEFINITION 5.4 We write $\Delta \vdash_S \gamma, \gamma' \twoheadrightarrow_{\mathrm{n}} \gamma''$ if for all $X \in \mathrm{dom}(\gamma)$ there is a K' such that $\Delta \vdash_S \gamma(X), \gamma'(X) \twoheadrightarrow_{\mathrm{n}} \gamma''(X) : K'$.

We first give some simple properties about the interpretation:

LEMMA 5.5 (*Basic Properties*)

1. If $[\![K]\!]\gamma\Delta$ is defined then there is a K' such that $\Delta \vdash_S K[\gamma] \twoheadrightarrow_{\mathrm{n}} K'$.
2. If $\gamma \in [\![\Gamma]\!]\Delta$ then $\Delta \vdash_S$ ok.
3. If $[\![K]\!]\gamma\Delta$ is defined and $\gamma'(X) = \gamma(X)$ for all $X \in \mathrm{dom}(\gamma)$ then $[\![K]\!]\gamma\Delta = [\![K]\!]\gamma'\Delta$.

4. If $\llbracket K \rrbracket \gamma \Delta$ and $\llbracket K \rrbracket \gamma' \Delta$ are defined and $\Delta \vdash_S \gamma, \gamma' \twoheadrightarrow_{\mathrm{n}} \gamma''$ then $\llbracket K \rrbracket \gamma \Delta = \llbracket K \rrbracket \gamma' \Delta$.

PROOF: Properties 1 and 2 follow by straightforward induction on K and Γ.

Property 3 follows by induction on K, using the fact that $A[\gamma] \equiv A[\gamma']$ for types A if $\gamma(X) = \gamma'(X)$ for all $X \in \mathrm{FV}(A)$, and similarly for kinds.

Property 4 follows by induction on K using the fact that $\Delta \vdash_S K[\gamma], K[\gamma'] \twoheadrightarrow_{\mathrm{n}} K''$ if there is a γ'' such that for all $X \in \mathrm{FV}(A)$ there is a K''' such that $\Delta \vdash_S \gamma(X), \gamma'(X) \twoheadrightarrow_{\mathrm{n}} \gamma''(X) : K'''$, which follows using Subject Reduction and Determinacy. □

Next, we need some properties similar to the usual saturated set conditions. In the following we assume $\Delta \vdash_S K[\gamma] \twoheadrightarrow_{\mathrm{n}} K'$:

LEMMA 5.6 (*Saturated Sets*)

1. If $A \in \llbracket K \rrbracket_{:} \gamma \Delta$ then A is a semantic object for Δ and K'.

2. If $(A, B) \in \llbracket K \rrbracket_{\leq} \gamma \Delta$ then $\Delta \vdash_S A \leq B : K'$.

3. If $(A, B) \in \llbracket K \rrbracket_{\leq} \gamma \Delta$ then $A \in \llbracket K \rrbracket_{:} \gamma \Delta$ and $B \in \llbracket K \rrbracket_{:} \gamma \Delta$.

4. If A and B are in $\llbracket K \rrbracket_{:} \gamma \Delta$ and $\Delta \vdash_S A, B \twoheadrightarrow_{\mathrm{n}} C : K'$ then $(A, B) \in \llbracket K \rrbracket_{\leq} \gamma \Delta$.

5. If (A, B) and (B, C) are in $\llbracket K \rrbracket_{\leq} \gamma \Delta$ then $(A, C) \in \llbracket K \rrbracket_{\leq} \gamma \Delta$.

6. If $\Delta(X)(A_1, \ldots, A_m) \in \llbracket K \rrbracket_{:} \gamma \Delta$ then $(X(A_1, \ldots, A_m), \Delta(X)(A_1, \ldots, A_m)) \in \llbracket K \rrbracket_{\leq} \gamma \Delta$.

7. If $\Delta \vdash_S A, B \twoheadrightarrow_{\mathrm{w}} C \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} D : K'$, $A \in \llbracket K \rrbracket_{:} \gamma \Delta$ and B is a semantic object for Δ and K' then $B \in \llbracket K \rrbracket_{:} \gamma \Delta$.

8. If $A \in \llbracket K \rrbracket_{:} \gamma \Delta$ then $(A, \mathrm{T}_{K[\gamma]}) \in \llbracket K \rrbracket_{\leq} \gamma \Delta$.

PROOF: Properties 1, 2 and 3 follow by construction.

Property 4 follows by induction on K, using Reflexivity and Conversion of subtyping in the semantics, plus Adequacy, Subject Reduction and Determinism for the Π case.

Property 5 follows by induction on K, using Transitivity of subtyping in the semantics.

For Property 6, we first show a lemma stating that if $\Delta(X)(A_1, \ldots, A_m)$ is a semantic object for Δ and K' then $\Delta \vdash_S X(A_1, \ldots, A_m) \leq \Delta(X)(A_1, \ldots, A_m) : K'$, which follows by definition of $\Delta(X)$, Adequacy and Reflexivity for subtyping in the semantics. Hence $X(A_1, \ldots, A_m)$ is a semantic object for Δ and K', using this lemma and Transitivity for the semantics. Property 6 then follows by induction on K, using Properties 1 and 3, and the fact that if $\vdash_S \Delta' \geq \Delta$ then $\Delta'(X) = \Delta(X)$ for all $X \in \mathrm{dom}(\Delta)$.

For Property 7, first notice that if $\Delta \vdash_S A, B \twoheadrightarrow_{\mathrm{w}} C \twoheadrightarrow_{\mathrm{n}} D : K'$ and A is a semantic object for Δ and K' then B is a semantic object for Δ and K' as well, using Determinism to show that $\Delta \vdash_S B \leq \mathrm{T}'_K : K'$. Furthermore, we can show that if $\Delta \vdash_S A, B \twoheadrightarrow_{\mathrm{w}} C \twoheadrightarrow_{\mathrm{n}} D : K$ and $\Delta \vdash_S A\,E \twoheadrightarrow_{\mathrm{w}} F \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} G : K'$ then $\Delta \vdash_S B\,E \twoheadrightarrow_{\mathrm{w}} F \;{}_{\mathrm{w}}\!\!\twoheadrightarrow_{\mathrm{n}} G : K'$. The result follows by induction on K, using Thinning and this simple lemma in the Π case.

Finally, for Property 8, we first observe that if $\Delta \vdash_S K \twoheadrightarrow_{\mathrm{n}} K'$ then T_K is a semantic object for Δ and K', which follows by a simple induction on K. The result follows by induction on K, using Properties 3, 4 and 5, plus Thinning and Conversion of subtyping. □

We also need properties corresponding to the model being a Kripke-model:

LEMMA 5.7 (*Monotonicity*) If $\vdash_S \Delta' \geq \Delta$ then:

1. $A \in [\![K]\!]_{:}\gamma\Delta$ implies $A \in [\![K]\!]_{:}\gamma\Delta'$.
2. $(A, B) \in [\![K]\!]_{:}\gamma\Delta$ implies $(A, B) \in [\![K]\!]_{\leq}\gamma\Delta'$.
3. $\gamma \in [\![\Gamma]\!]\Delta$ implies $\gamma \in [\![\Gamma]\!]\Delta'$.

PROOF: By induction on K or Γ, using Thinning for the first two and using the first two for the last. □

We also need to account for the dependency, since bounds in kinds include types:

LEMMA 5.8 (*Substitution*)

1. $[\![K]\!]\gamma_1[X{:=}B[\gamma_1]]\gamma_2\Delta = [\![K[X{\leftarrow}B]]\!]\gamma_1\gamma_2$.
2. Suppose that:
   - $[\![\Gamma_1,\ X{\leq}B{:}K,\ \Gamma_2]\!]\Delta$ is defined, and
   - $(A[\gamma_1], B[\gamma_1]) \in [\![K]\!]\gamma_1\Delta$ for $\gamma_1 \in [\![\Gamma_1]\!]\Delta$.

   Then:

   - $[\![\Gamma_1,\ \Gamma_2[X{\leftarrow}A]]\!]\Delta$ is defined, and
   - if $\gamma_1\gamma_2 \in [\![\Gamma_1,\ \Gamma_2[X{\leftarrow}A]]\!]\Delta$ then $\gamma_1[X{:=}A[\gamma_1]]\gamma_2 \in [\![\Gamma_1,\ X{\leq}B{:}K,\ \Gamma_2]\!]\Delta$.

PROOF: Case 1 follows by induction on K, using basic properties of parallel substitution. Case 2 follows by induction on Γ_2, using basic properties of parallel substitution and Case 1. □

Finally, we prove a lemma to deal with the rules of context equality.

LEMMA 5.9 (*Context Replacement*) Suppose:

- $[\![\Gamma_1,\ X{\leq}A{:}K,\ \Gamma_2]\!]\Delta$ is defined, and
- for any $\gamma_1 \in [\![\Gamma_1]\!]\Delta$ then:
  - $\Delta \vdash_S K[\gamma_1], K'[\gamma_1] \twoheadrightarrow_{\mathrm{n}} K''$ and $\Delta \vdash_S A[\gamma_1], B[\gamma_1] \twoheadrightarrow_{\mathrm{n}} C : K''$,
  - $[\![K]\!]\gamma_1\Delta = [\![K']\!]\gamma_1\Delta$, and
  - $A[\gamma_1] \in [\![K]\!]_{:}\gamma_1\Delta$ and $B[\gamma_1] \in [\![K']\!]_{:}\gamma_1\Delta$.

Then $[\![\Gamma_1,\ X{\leq}A{:}K,\ \Gamma_2]\!]\Delta = [\![\Gamma_1,\ X{\leq}B{:}K',\ \Gamma_2]\!]\Delta$.

PROOF: By induction on Γ_1, using Lemma 5.6 Case 4 for the base case. □

5.3 Soundness

We can now prove soundness. As usual for strong normalization proofs, we first need to prove the more general statement with respect to arbitrary well-behaved substitutions.

THEOREM 5.10 If $\Gamma \vdash J$ and $\Delta \vdash_S$ ok then $[\![\Gamma]\!]\Delta$ is defined. Furthermore:

1. If $\Gamma \vdash K$ and $\gamma \in [\![\Gamma]\!]\Delta$ then $[\![K]\!]\gamma\Delta$ is defined.
2. If $\Gamma \vdash A : K$ and $\gamma \in [\![\Gamma]\!]\Delta$ then $A[\gamma] \in [\![K]\!]_{:}\gamma\Delta$.
3. If $\Gamma \vdash K = K'$ and $\gamma \in [\![\Gamma]\!]\Delta$ then $[\![K]\!]\gamma\Delta = [\![K']\!]\gamma\Delta$ and there is a K'' such that $\Delta \vdash_S K[\gamma], K'[\gamma] \twoheadrightarrow_{\mathrm{n}} K''$.
4. If $\Gamma \vdash A = B : K$ and $\gamma \in [\![\Gamma]\!]\Delta$ then $A[\gamma]$ and $B[\gamma]$ are in $[\![K]\!]_{\leq}\gamma\Delta$ and there are C and K' such that $\Delta \vdash_S K[\gamma] \twoheadrightarrow_{\mathrm{n}} K'$ and $\Delta \vdash_S A[\gamma], B[\gamma] \twoheadrightarrow_{\mathrm{n}} C : K'$.
5. If $\Gamma \vdash A \leq B : K$ and $\gamma \in [\![\Gamma]\!]\Delta$ then $(A[\gamma], B[\gamma]) \in [\![K]\!]_{\leq}\gamma\Delta$.

PROOF: By induction on derivations, using the above properties about the interpretation. We consider several cases:

C-EMPTY By definition $[\![\emptyset]\!]\Delta$ is defined if $\Delta \vdash_S$ ok.

C-VAR By inductive hypothesis $[\![\Gamma]\!]\Delta'$ is defined for $\vdash_S \Delta' \geq \Delta$. If $\gamma \in [\![\Gamma]\!]\Delta'$, then $A[\gamma] \in [\![\star]\!]_{:}\gamma\Delta'$ by inductive hypothesis. Furthermore, $x \notin \mathrm{dom}(\Gamma)$, so $[\![\Gamma,\ x{:}A]\!]\Delta$ is defined.

C-TVAR By inductive hypothesis $[\![\Gamma]\!]\Delta'$ is defined for $\vdash_S \Delta' \geq \Delta$. If $\gamma \in [\![\Gamma]\!]\Delta'$, then $A[\gamma] \in [\![K]\!]_{:}\gamma\Delta'$ by inductive hypothesis, and $X \notin \mathrm{dom}(\Gamma)$, so $[\![\Gamma,\ X{\leq}A{:}K]\!]\Delta$ is defined.

KIND AGREEMENT By inductive hypothesis $[\![\Gamma]\!]\Delta$ is defined.

Suppose $\gamma \in [\![\Gamma]\!]\Delta$. By inductive hypothesis $A[\gamma] \in [\![K]\!]_{:}\gamma\Delta$, so $[\![K]\!]_{:}\gamma\Delta$ is defined. Furthermore, by Lemma 5.5 Case 1 there is a K' such that $\Delta \vdash_S K[\gamma] \twoheadrightarrow_{\mathrm{n}} K'$.

K-SUBST By inductive hypothesis $[\![\Gamma_1,\ X{\leq}B{:}K,\ \Gamma_2]\!]\Delta$ and $[\![\Gamma_0]\!]\Delta$ are defined, and $\gamma_0 \in [\![\Gamma_0]\!]\Delta$ implies $(A[\gamma_0], B[\gamma_0]) \in [\![K]\!]_{\leq}\gamma_0\Delta$. By Lemma 5.8 Case 2 $[\![\Gamma_1,\ \Gamma_2[X{\leftarrow}A]]\!]\Delta$ is defined.

Furthermore, $\gamma_0\gamma_1 \in [\![\Gamma_1,\ \Gamma_2[X{\leftarrow}A]]\!]\Delta$ implies $\gamma_0[X{:=}A[\gamma_0]]\gamma_1 \in [\![\Gamma_1,\ X{\leq}A{:}K',\ \Gamma_2]\!]\Delta$, so $[\![K]\!]\gamma_0[X{:=}A[\gamma_0]]\gamma_1\Delta$ is defined by inductive hypothesis, and $[\![K]\!]\gamma_0[X{:=}A[\gamma_0]]\gamma_1\Delta = [\![K[X{\leftarrow}A]]\!]\gamma_0\gamma_1\Delta$ by Lemma 5.8 Case 1.

K-CTXT-EQ The inductive hypotheses satisfy the premises of Lemma 5.9, so

$$[\![\Gamma_1,\ X{\leq}A{:}K,\ \Gamma_2]\!]\Delta \quad = \quad [\![\Gamma_1,\ X{\leq}B{:}K',\ \Gamma_2]\!]\Delta$$

Hence, $[\![\Gamma_1,\ X{\leq}B{:}K',\ \Gamma_2]\!]\Delta$ is defined, and if $\gamma \in [\![\Gamma_1,\ X{\leq}B{:}K',\ \Gamma_2]\!]\Delta$ then $\gamma \in [\![\Gamma_1,\ X{\leq}A{:}K,\ \Gamma_2]\!]\Delta$, so by inductive hypothesis $[\![K]\!]\gamma\Delta$ is defined.

T-Top By inductive hypothesis $[\![\Gamma]\!]\Delta$ is defined.

Suppose $\gamma \in [\![\Gamma]\!]\Delta$. We know $\Delta \vdash_S \mathsf{ok}$ by Lemma 5.5 Case 2. Hence $\Delta \vdash_S \mathrm{T}_\star \twoheadrightarrow_{\mathrm{n}} \mathrm{T}_\star : \star$ by ST-Top, and $\Delta \vdash_S \mathrm{T}_\star \leq_W \mathrm{T}_\star : \star$ by SWS-Top and SS-Inc. Hence $\mathrm{T}_\star$ is a semantic object for Δ and $\star$, and so $\mathrm{T}_\star \in [\![\star]\!]_{:}\gamma\Delta$.

T-TVar By inductive hypothesis $[\![\Gamma_1, X \leq A : K, \Gamma_2]\!]\Delta$ is defined.

Suppose $\gamma \in [\![\Gamma]\!]\Delta$. Then $\gamma = \gamma_1[X{:=}B]\gamma_2$, with $\gamma_1 \in [\![\Gamma_1]\!]\Delta$ and $(B, A[\gamma_1]) \in [\![K]\!]_{\leq}\gamma_1\Delta$, by definition of $[\![\Gamma]\!]\Delta$. Hence $B \in [\![K]\!]_{:}\gamma_1\Delta$ by Lemma 5.6 Case 3, and $B \in [\![K]\!]_{:}\gamma\Delta$ by Lemma 5.5 Case 3.

T-TAbs By inductive hypothesis $[\![\Gamma,\ X{\leq}A_1{:}K_1]\!]\Delta$ is defined, so by definition of the interpretation $[\![\Gamma]\!]\Delta'$ is defined and $A_1[\gamma] \in [\![K_1]\!]_{:}\gamma\Delta'$ for $\gamma \in [\![\Gamma]\!]\Delta'$, for any $\vdash_S \Delta' \geq \Delta$.

Suppose $\gamma \in [\![\Gamma]\!]\Delta$. We want to show $(\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma] \in [\![\Pi X{\leq}A_1{:}K_1.K_2]\!]_{:}\gamma\Delta$. We have to show two conditions:

- $\Delta \vdash_S (\Pi X{\leq}A_1{:}K_1.K_2)[\gamma] \twoheadrightarrow_{\mathrm{n}} K'$ and $(\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma]$ is a semantic object for Δ and K'. By Lemma 5.5 Case 1 $\Delta \vdash_S K_1[\gamma] \twoheadrightarrow_{\mathrm{n}} K_1'$, and by Lemma 5.6 Case 1 $\Delta \vdash_S A_1[\gamma] \twoheadrightarrow_{\mathrm{n}} A_1' : K_1'$. Hence $\Delta,\ Y{\leq}A_1[\gamma]{:}K_1[\gamma] \vdash_S \mathsf{ok}$ for Y fresh in Δ.

  By Lemma 5.7 Case 3 $\gamma \in [\![\Gamma]\!]\Delta$ implies $\gamma \in [\![\Gamma]\!]\Delta,\ Y{\leq}A_1[\gamma]{:}K_1[\gamma]$. By Lemma 5.6 Case 6 $(Y, A_1[\gamma]) \in [\![K_1]\!]_{\leq}\gamma\Delta,\ Y{\leq}A_1[\gamma]{:}K_1[\gamma]$. Hence

  $$\gamma[X{:=}Y] \in [\![\Gamma,\ X{\leq}A_1{:}K_1]\!]\Delta,\ Y{\leq}A_1[\gamma]{:}K_1[\gamma]$$

  so by inductive hypothesis $A_2[\gamma[X{:=}Y]] \in [\![K_2]\!]_{:}\gamma[X{:=}Y]\Delta'$. Hence

  $$\begin{array}{lll} \Delta,\ Y{\leq}A_1[\gamma]{:}K_1[\gamma] & \vdash_S & K_2[\gamma[X{:=}Y]] \twoheadrightarrow_{\mathrm{n}} K_2' \\ \Delta,\ Y{\leq}A_1[\gamma]{:}K_1[\gamma] & \vdash_S & A_2[\gamma[X{:=}Y]] \twoheadrightarrow_{\mathrm{n}} A_2' : K_2' \\ \Delta,\ Y{\leq}A_1[\gamma]{:}K_1[\gamma] & \vdash_S & A_2[\gamma[X{:=}Y]] \leq \mathrm{T}_{K_2'} : K_2' \end{array}$$

  by Lemma 5.6 Case 1 and definition of semantic object. Hence,

  $$\begin{array}{lll} \Delta & \vdash_S & (\Pi X{\leq}A_1{:}K_1.K_2)[\gamma] \twoheadrightarrow_{\mathrm{n}} \Pi Y{\leq}A_1'{:}K_1'.K_2' \\ \Delta & \vdash_S & (\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma] \equiv \Lambda Y{\leq}A_1[\gamma]{:}K_1[\gamma].A_2[\gamma[X{:=}Y]] \\ & \twoheadrightarrow_{\mathrm{n}} & \Lambda Y{\leq}A_1'{:}K_1'.A_2' : \Pi Y{\leq}A_1'{:}K_1'.K_2' \\ \Delta & \vdash_S & (\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma] \leq \Lambda Y{\leq}A_1'{:}K_1'.\mathrm{T}_{K_2'} \equiv \mathrm{T}_{\Pi Y\leq A_1':K_1'.K_2'} \\ & : & \Pi Y{\leq}A_1'{:}K_1'.K_2' \end{array}$$

  where the last line follows using Adequacy and Subject Reduction for the well-typedness of the right-hand side. Hence, $(\Pi X{\leq}A_1{:}K_1.K_2)[\gamma]$ is a semantic object for Δ and $\Pi Y{\leq}A_1'{:}K_1'.K_2'$.

- $(\Pi X{\leq}A_1{:}K_1.K_2)[\gamma]\, B \in [\![K_2]\!]_{:}\gamma[X{:=}B]\Delta'$, with $\vdash_S \Delta' \geq \Delta$ and $(B, A_1[\gamma]) \in [\![K_1]\!]_{\leq}\gamma\Delta'$. First, by Lemma 5.7 Case 3 $\gamma \in [\![\Gamma]\!]\Delta$ implies $\gamma \in [\![\Gamma]\!]\Delta'$, and $(B, A_1[\gamma]) \in [\![K_1]\!]_{\leq}\gamma\Delta'$ implies $\gamma[X{:=}B] \in [\![\Gamma,\ X{\leq}A_1{:}K_1]\!]\Delta'$ by definition of the interpretation. Hence by inductive hypothesis $A_2[\gamma[X{:=}B]] \in [\![K_2]\!]_{:}\gamma[X{:=}B]\Delta'$, and so by Lemma 5.6 Case 1 $\Delta' \vdash_S A_2[\gamma[X{:=}B]] \twoheadrightarrow_{\mathrm{w}} C : K_2''$, where $\Delta' \vdash_S K_2[\gamma[X{:=}B]] \twoheadrightarrow_{\mathrm{n}} K_2''$. Then $\Delta \vdash_S (\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma] : \Pi Y{\leq}A_1'{:}K_1'.K_2'$ implies $\Delta' \vdash_S (\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma] : \Pi Y{\leq}A_1'{:}K_1'.K_2'$ by Thinning, and $\Delta' \vdash_S B \leq A_1[\gamma] : K_1'$ by Lemma 5.6 Case 2. Hence

$$\begin{array}{rcl} \Delta' & \vdash_S & (\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma](B) \\ & \equiv & (\Lambda Y{\leq}A_1[\gamma]{:}K_1[\gamma].A_2[\gamma[X{:=}Y]])(B) \twoheadrightarrow_{\mathrm{w}} C : K_2'' \end{array}$$

by ST-Beta, because $A_2[\gamma[X{:=}Y]][Y{\leftarrow}B] \equiv A_2[\gamma[X{:=}B]]$ where Y can be chosen to be fresh in Δ'. Hence, by Lemma 5.6 Case 7

$$(\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma](B) \in [\![K_2]\!]_{:}\gamma[X{:=}B]\Delta'$$

Hence, $(\Lambda X{\leq}A_1{:}K_1.A_2)[\gamma](B) \in [\![\Pi X{\leq}A_1{:}K_1.K_2]\!]_{:}\gamma\Delta$.

T-TApp By inductive hypothesis $[\![\Gamma]\!]\Delta$ is defined.

Suppose $\gamma \in [\![\Gamma]\!]\Delta$. Then $A[\gamma] \in [\![\Pi X{\leq}B{:}K_1.K_2]\!]_{:}\gamma\Delta$ and $(C[\gamma], B[\gamma]) \in [\![K_1]\!]_{\leq}\gamma\Delta$. Clearly $\vdash_S \Delta \geq \Delta$, since $\Delta \vdash_S \mathrm{ok}$ by Lemma 5.5 Case 2, so

$$(A[\gamma])(C[\gamma]) \equiv (A\,C)[\gamma] \in [\![K_2]\!]_{:}\gamma[X{:=}C[\gamma]] = [\![K_2[X{\leftarrow}C]]\!]_{:}\gamma\Delta$$

where the last equality follows by Lemma 5.8 Case 2.

T-Eq-TApp By inductive hypothesis $[\![\Gamma]\!]\Delta$ is defined.

Suppose $\gamma \in [\![\Gamma]\!]\Delta$. By inductive hypothesis $A[\gamma], B[\gamma] \in [\![\Pi X{\leq}E{:}K_1.K_2]\!]_{:}\gamma\Delta$, $\Delta \vdash_S A[\gamma], B[\gamma] \twoheadrightarrow_{\mathrm{n}} A' : \Pi X{\leq}E'{:}K_1'.K_2'$ with $\Delta \vdash_S (\Pi X{\leq}E{:}K_1.K_2)[\gamma] \twoheadrightarrow_{\mathrm{n}} \Pi X{\leq}E'{:}K_1'.K_2'$, and also $C[\gamma], D[\gamma] \in [\![K_1]\!]_{:}\gamma\Delta$, $\Delta \vdash_S C[\gamma], D[\gamma] \twoheadrightarrow_{\mathrm{n}} C' : K_1'$, and finally $(C[\gamma], E[\gamma]) \in [\![K_1]\!]_{\leq}\gamma\Delta$. Hence,

$$\begin{array}{rcl} (A[\gamma])(C[\gamma]) \equiv (AC)[\gamma] & \in & [\![K_2]\!]_{:}\gamma[X{:=}C[\gamma]]\Delta = [\![K_2[X{\leftarrow}C]]\!]_{:}\gamma\Delta \\ (B[\gamma])(D[\gamma]) \equiv (BD)[\gamma] & \in & [\![K_2]\!]_{:}\gamma[X{:=}D[\gamma]]\Delta = [\![K_2[X{\leftarrow}D]]\!]_{:}\gamma\Delta \end{array}$$

Using Adequacy, Subject Reduction and Determinacy we conclude that

$$\Delta \vdash_S (AC)[\gamma], (BD)[\gamma] \twoheadrightarrow_{\mathrm{n}} F : K_2'$$

where $\Delta \vdash_S K_2[X{\leftarrow}C] \twoheadrightarrow_{\mathrm{n}} K_2'$. Finally, $\Delta \vdash_S \gamma[X{:=}C[\gamma]], \gamma[X{:=}D[\gamma]] \twoheadrightarrow_{\mathrm{n}} \gamma'$, so by Lemma 5.5 Case 4:

$$(BD)[\gamma] \in [\![K_2]\!]_{:}\gamma[X{:=}C[\gamma]]\Delta = [\![K_2[X{\leftarrow}C]]\!]_{:}\gamma\Delta$$

S-Top By inductive hypothesis $[\![\Gamma]\!]\Delta$ is defined.

Suppose $\gamma \in [\![\Gamma]\!]\Delta$. Then $A[\gamma] \in [\![K]\!]_{:}\gamma\Delta$, so by Lemma 5.6 Case 8 $(A[\gamma], \mathrm{T}_{K[\gamma]}) \in [\![K]\!]_{\leq}\gamma\Delta$, and $\mathrm{T}_{K[\gamma]} \equiv \mathrm{T}_K[\gamma]$.

S-Trans By inductive hypothesis $[\![\Gamma]\!]\Delta$ is defined.

Suppose $\gamma \in [\![\Gamma]\!]\Delta$. By inductive hypothesis $(A[\gamma], B[\gamma]) \in [\![K]\!]_{\leq}\gamma\Delta$ and $(B[\gamma], C[\gamma]) \in [\![K]\!]_{\leq}\gamma\Delta$. Hence by Lemma 5.6 Case 5 $(A[\gamma], C[\gamma]) \in [\![K]\!]_{\leq}\gamma\Delta$. □

Lemma 5.11 If $\Gamma \vdash \text{ok}$ then $\text{id}_\Gamma \in [\![\Gamma]\!]\Gamma$, where id_Γ is the identity substitution on Γ.

Proof: By induction on Γ, using several simple generation lemmas for contexts. □

Corollary 5.12 (*Soundness*)

1. If $\Gamma \vdash A : K$ then there are K', B and C such that $\Gamma \vdash_S K \twoheadrightarrow_{\text{n}} K'$ and $\Gamma \vdash_S A \twoheadrightarrow_{\text{w}} B \;{}_{\text{w}}\!\!\twoheadrightarrow_{\text{n}} C : K'$.

2. If $\Gamma \vdash A \leq B : K$ then there is a K' such that $\Gamma \vdash_S K \twoheadrightarrow_{\text{n}} K'$ and $\Gamma \vdash_S A \leq B : K'$.

5.4 Consequences of Soundness

We can use Soundness, Corollary 5.12, and Completeness, Proposition 4.6, to transfer the metatheoretic results from the typed operational semantics to the original presentation.

Lemma 5.13 (*Admissibility of Structural Rules*) The rules in Section 2.3 are admissible for the system $\Gamma \vdash^- J$.

Proof: Suppose we have a derivation of $\Gamma \vdash J$, for example $\Gamma \vdash A : K$, which is then a derivation of $\Gamma \vdash^- J$ with uses of the structural rules in Section 2.3. By Soundness we know that there are B and K' such that $\Gamma \vdash_S A \twoheadrightarrow_{\text{n}} B : K'$ and $\Gamma \vdash_S K \twoheadrightarrow_{\text{n}} K'$. By Completeness $\Gamma \vdash^- A : K'$ and $\Gamma \vdash^- K = K'$, so by T-Conv $\Gamma \vdash^- A : K$. □

Lemma 5.14 (*Strong Normalization*) If $\Gamma \vdash A : K$ then A is strongly normalizing.

Proof: By Soundness and Strong Normalization (Lemma 4.15). □

Lemma 5.15 (*Subject Reduction for $\to_{\beta_2}$*)

- If $\Gamma \vdash \text{ok}$ and $\Gamma \to_{\beta_2} \Gamma'$ then $\Gamma' \vdash \text{ok}$.
- If $\Gamma \vdash K$ and $K \to_{\beta_2} K'$ then $\Gamma \vdash K$ and $\Gamma \vdash K =_\beta K'$.
- If $\Gamma \vdash A : K$ and $A \to_{\beta_2} B$ then $\Gamma \vdash A : K$ and $\Gamma \vdash A =_\beta B : K$.
- If $\Gamma \vdash A \leq B : K$ and $A \to_{\beta_2} C$ then $\Gamma \vdash C \leq B : K$, or if $B \to_{\beta_2} C$ then $\Gamma \vdash A \leq C : K$.

Proof: By Soundness, Subject Reduction (Corollary 4.13), and Completeness. □

Proposition 5.16 (*Generation for Subtyping*)

1. If $\Gamma \vdash (A_1 \rightarrow A_2) \leq (B_1 \rightarrow B_2) : \star$ then $\Gamma \vdash B_1 \leq A_1 : \star$ and $\Gamma \vdash A_2 \leq B_2 : \star$
2. If $\Gamma \vdash (\forall X{\leq}A_1{:}K_A.A_2) \leq (\forall X{\leq}B_1{:}K_B.B_2) : \star$ then $\Gamma \vdash K_A =_\beta K_B$, $\Gamma \vdash A_1 =_\beta B_1 : K_A$, and $\Gamma, X{\leq}A_1{:}K_A \vdash A_2 \leq B_2 : \star$.

PROOF:

1. By Soundness $\Gamma \vdash_S A_1 \rightarrow A_2 \leq B_1 \rightarrow B_2 : \star$. Since the semantic presentation is deterministic the latter must have been obtained by SS-INC and SWS-ARROW from $\Gamma \vdash_S B_1 \leq A_1 : \star$ and $\Gamma \vdash_S A_2 \leq B_2 : \star$. Then, by Completeness, $\Gamma \vdash B_1 \leq A_1 : \star$ and $\Gamma \vdash A_2 \leq B_2 : \star$.
2. By Soundness $\Gamma \vdash_S \forall X{\leq}A_1{:}K_A.A_2 \leq \forall X{\leq}B_1{:}K_B.B_2 : \star$. Since the semantic presentation is deterministic the latter must have been obtained by SS-INC and SWS-ALL from $\Gamma, X{\leq}A_1{:}K_A \vdash_S A_2 \leq B_2 : \star$, $\Gamma \vdash_S A_1, B_1 \twoheadrightarrow_{\mathrm{n}} C : K''$, and $\Gamma \vdash_S K_A, K_B \twoheadrightarrow_{\mathrm{n}} K''$. By Completeness $\Gamma, X{\leq}A_1{:}K_A \vdash A_2 \leq B_2 : \star$, and also $\Gamma \vdash A_1 =_\beta C : K''$, $\Gamma \vdash B_1 =_\beta C : K''$, $\Gamma \vdash K_A =_\beta K''$, and $\Gamma \vdash K_B =_\beta K''$. Hence by T-EQ-SYM, T-EQ-TRANS, K-EQ-SYM, K-EQ-TRANS, it follows that $\Gamma \vdash K_A =_\beta K_B$, and also that $\Gamma \vdash A_1 =_\beta B_1 : K''$, and $\Gamma \vdash K'' =_\beta K_A$. Finally, by T-EQ-CONV, $\Gamma \vdash A_1 =_\beta B_1 : K_A$. □

PROPOSITION 5.17 (*Generation for Typing*)

1. If $\Gamma \vdash \lambda x{:}A_1.M : A$ then there exists an A_2 such that $\Gamma, x{:}A_1 \vdash M : A_2$ and $\Gamma \vdash A_1 \rightarrow A_2 \leq A : \star$.
2. If $\Gamma \vdash \lambda X{\leq}A_1{:}K.M : A$ then there exists an A_2 such that $\Gamma, X{\leq}A_1{:}K \vdash M : A_2$ and $\Gamma \vdash \forall X{\leq}A_1{:}K.A_2 \leq A : \star$.

PROOF: Each case follows by induction on the derivation of the antecedent. □

LEMMA 5.18 (*Agreement*)

1. If $\Gamma_1, x{:}A, \Gamma_2 \vdash \mathrm{ok}$ then $\Gamma_1 \vdash A : \star$.
2. If $\Gamma_1, X{\leq}A{:}K, \Gamma_2 \vdash \mathrm{ok}$ then $\Gamma_1 \vdash A : K$.
3. If $\Gamma \vdash M : A$ then $\Gamma \vdash A : \star$.

PROOF: We use Lemma 4.23, Soundness and Completeness. □

6 Subject Reduction

PROPOSITION 6.1 ($\rightarrow_{\beta_1}$ *Subject Reduction for Terms*)
If $\Gamma \vdash M : A$ and $M \rightarrow_{\beta_1} M'$ then $\Gamma \vdash M' : A$.

PROOF: By induction on the derivation of $\Gamma \vdash M : A$.

T-VAR Vacuously true.

T-ABS By the induction hypothesis and T-ABS.

T-APP Let $M \equiv N\,a$. Then we are given that $\Gamma \vdash N : A_1 \rightarrow A$ and $\Gamma \vdash a : A_1$. There are 3 cases to consider. If $N \rightarrow_{\beta_1} N'$ or $a \rightarrow_{\beta_1} a'$ then the result follows by the induction hypothesis and T-APP. The interesting case is when $N \equiv \lambda x{:}B_1.N'$ and $M \rightarrow_{\beta_1} N'[x \leftarrow a]$. By Generation for Typing (Proposition 5.17), $\Gamma, x{:}B_1 \vdash N' : B_2$ and $\Gamma \vdash B_1 \rightarrow B_2 \leq A_1 \rightarrow A$, for some B_2. By Generation for Subtyping (Proposition 5.16) $\Gamma \vdash A_1 \leq B_1$ and $\Gamma \vdash B_2 \leq A$. By T-SUB, $\Gamma \vdash a : B_1$, and, by Substitution (Lemma 4.26), $\Gamma \vdash N'[x \leftarrow a] : B_2$. Finally, by T-SUB, $\Gamma \vdash N'[x \leftarrow a] : A$.

T-TABS By the induction hypothesis and T-TABS.

T-TAPP Let $M \equiv N\,B$. We are given that $\Gamma \vdash N : \forall X{\leq}A_1{:}K_A.A_2$, $\Gamma \vdash B \leq A_1 : K_A$, and $A \equiv A_2[X \leftarrow B]$. There are two cases to consider. If $N \rightarrow_{\beta_1} N'$ then the result follows by the induction hypothesis and T-TAPP. Otherwise, $N \equiv \lambda X{\leq}B_1{:}K_B.N'$ and $M' \equiv N'[X \leftarrow B]$. By Generation for Typing (Proposition 5.17), $\Gamma,\ X{\leq}B_1{:}K_B \vdash N' : B_2$, and $\Gamma \vdash \forall X{\leq}B_1{:}K_B.B_2 \leq \forall X{\leq}A_1{:}K_A.A_2 : \star$, for some B_2. By Generation for Subtyping (Proposition 5.16), $\Gamma,\ X{\leq}A_1{:}K_A \vdash B_2 \leq A_2 : \star$, $\Gamma \vdash A_1 =_\beta B_1 : K_A$, and $\Gamma \vdash K_A =_\beta K_B$. By S-CONV, $\Gamma \vdash A_1 \leq B_1 : K_A$, and, by S-TRANS, $\Gamma \vdash B \leq B_1 : K_A$. By Substitution (Lemma 4.26), $\Gamma \vdash N'[X \leftarrow B] : B_2[X \leftarrow B]$. By SUBST, $\Gamma \vdash B_2[X \leftarrow B] \leq A_2[X \leftarrow B] : \star$. Finally, by T-SUB $\Gamma \vdash N'[X \leftarrow B] : A_2[X \leftarrow B] : \star$.

T-SUB By the induction hypothesis and T-SUB. □

LEMMA 6.2 If $\Gamma \vdash M : A$ and $\Gamma \rightarrow_{\beta_2} \Gamma'$ then $\Gamma' \vdash M : A$.

PROOF: By induction on derivations using Lemma 5.15. □

LEMMA 6.3 If $C \rightarrow_{\beta_2} C'$ then $B[X \leftarrow C] \twoheadrightarrow_{\beta_2} B[X \leftarrow C']$.

PROPOSITION 6.4 ($\rightarrow_{\beta_2}$ *Subject Reduction for Terms*)
If $\Gamma \vdash M : A$ and $M \rightarrow_{\beta_2} M'$ then $\Gamma \vdash M' : A$.

PROOF: By induction on derivations.

T-VAR Vacuously true.

T-ABS There are two cases to consider.

1. $A \to_{\beta_2} A'$. By Lemma 6.2, $\Gamma, x{:}A' \vdash M : B$, by T-ABS, $\Gamma \vdash \lambda x{:}A'.M : A' \to B$. By Lemma 5.18 Case 1 and Lemma 4.23 we conclude $\Gamma \vdash A : \star$ from $\Gamma,\ x{:}A \vdash M : B$. By Lemma 5.15, $\Gamma \vdash A =_\beta A' : \star$. By Lemma 5.18 Case 3, $\Gamma,\ x{:}A' \vdash B : \star$, by Strengthening (Lemma 4.24), $\Gamma \vdash B : \star$, by T-EQ-REFL, $\Gamma \vdash B =_\beta B : \star$, by T-EQ-ARROW and T-EQ-SYM, $\Gamma \vdash A' \to B =_\beta A \to B : \star$. Then, by S-CONV, $\Gamma \vdash A' \to B \leq A \to B : \star$, and by T-SUB $\Gamma \vdash \lambda x{:}A'.M : A \to B$
2. $M \to_{\beta_2} M'$. By the induction hypothesis and T-ABS.

T-APP There are two cases to consider, either $M \to_{\beta_2} M'$ or $N \to_{\beta_2} N'$. Both cases follow by the induction hypothesis and T-APP.

T-TABS There are three cases to consider.

1. If $M \to_{\beta_2} M'$, the result follows by the induction hypothesis and T-TABS.
2. If $A \to_{\beta_2} A'$, it is similar to T-ABS case 1.
3. If $K \to_{\beta_2} K'$, by Lemma 6.2, $\Gamma,\ X{\leq}A{:}K' \vdash M : B$, by T-TABS, $\Gamma \vdash \lambda X{\leq}A{:}K'.M : \forall X{\leq}A{:}K'.B$. From $\Gamma,\ X{\leq}A{:}K \vdash M : B$, by Lemma 4.23 and Lemma 5.18 Case 2, $\Gamma \vdash A : K$, and, by the rule KIND-AGREEMENT, $\Gamma \vdash K$. By Lemma 5.15, $\Gamma \vdash K =_\beta K'$, by T-EQ-REFL, $\Gamma \vdash A =_\beta A : K$. With a similar argument to that used in the T-ABS case we prove that $\Gamma \vdash B =_\beta B : \star$. Now, by T-EQ-SYM and T-EQ-ALL, $\Gamma \vdash \forall X{\leq}A{:}K'.B =_\beta \forall X{\leq}A{:}K.B : \star$, by S-CONV, $\Gamma \vdash \forall X{\leq}A{:}K'.B \leq \forall X{\leq}A{:}K.B : \star$. Finally, by T-SUB, $\Gamma \vdash \lambda X{\leq}A{:}K'.M : \forall X{\leq}A{:}K.B$.

T-TAPP There are two cases to consider.

1. If $M \to_{\beta_2} M'$ then the result follows by the induction hypothesis and T-TAPP.
2. If $C \to_{\beta_2} C'$, by Lemma 4.13, $\Gamma \vdash C' \leq A : K$, by T-TAPP, $\Gamma \vdash M\,C' : B[X{\leftarrow}C']$, by Lemma 5.18 Case 3, $\Gamma \vdash B[X{\leftarrow}C] : \star$, by Lemma 6.3 and Lemma 5.15, $\Gamma \vdash B[X{\leftarrow}C] =_\beta B[X{\leftarrow}C'] : \star$, by T-EQ-SYM and T-SUB, $\Gamma \vdash M\,C' : B[X{\leftarrow}C]$.

T-SUB By the induction hypothesis and T-SUB. □

PROPOSITION 6.5 ($\twoheadrightarrow_\beta$ *Subject Reduction for Terms*)
If $\Gamma \vdash M : A$ and $M \twoheadrightarrow_\beta M'$ then $\Gamma \vdash M' : A$.

PROOF: The proof is by induction on the definition of $\twoheadrightarrow_\beta$. The reflexive case is immediate, the cases for $\to_{\beta_1}$ and $\to_{\beta_2}$ follow by Propositions 6.1 and 6.4 respectively, and the transitivity case follows by the induction hypothesis. □

7 An Algorithmic Presentation

From the rules for subtyping in the typed operational semantics we extract an algorithm that computes the subtyping relation on weak head normal forms, ignoring kind information and well-formation of contexts. The reductions to weak head normal form are then untyped calculations defined as follows:

$$\begin{array}{rcll} A & \twoheadrightarrow_{\mathrm{w}} & A & \text{if } A \text{ is weak head normal.} \\ A\,B & \twoheadrightarrow_{\mathrm{w}} & C\,D & \text{if } A \twoheadrightarrow_{\mathrm{w}} C,\ B \twoheadrightarrow_{\mathrm{n}} D, \text{ and } C \neq \Lambda X{\leq}E{:}K.F \\ A\,B & \twoheadrightarrow_{\mathrm{w}} & E & \text{if } A \twoheadrightarrow_{\mathrm{w}} \Lambda X{\leq}C{:}K.D \text{ and } D[X{\leftarrow}B] \twoheadrightarrow_{\mathrm{w}} E \end{array}$$

where $\twoheadrightarrow_{\mathrm{n}}$ is defined simultaneously by recursively finding the normal form of each subterm in a weak head normal form.

The algorithm to compute the subtyping relation is defined by the following rules.

7.1 Algorithmic Subtyping

$$\frac{A \twoheadrightarrow_{\mathrm{w}} C \qquad B \twoheadrightarrow_{\mathrm{w}} D \qquad \Gamma \vdash_A C \leq_W D}{\Gamma \vdash_A A \leq B} \quad (\textsc{AS-Inc})$$

This rule reduces the arguments to weak head normal and then invokes the weak-head subtyping algorithm defined as follows.

7.2 Algorithmic Weak-Head Subtyping

$$\Gamma \vdash_A A \leq_W \mathrm{T}_\star \qquad A \text{ does not have a head variable} \quad (\textsc{AWS-Top})$$

$$\frac{\begin{array}{ll} \Gamma(X) \twoheadrightarrow_{\mathrm{n}} B & \Gamma \vdash_A E \leq_W C \\ B(A_1,\ldots,A_m) \twoheadrightarrow_{\mathrm{w}} E & C \neq X(A_1,\ldots A_n) \end{array}}{\Gamma \vdash_A X(A_1,\ldots,A_m) \leq_W C : K} \quad (\textsc{AWS-TApp})$$

$$\Gamma \vdash_A X(A_1,\ldots,A_m) \leq_W X(A_1,\ldots,A_m) \quad (\textsc{AWS-Refl})$$

$$\frac{\Gamma \vdash_A B_1 \leq A_1 \qquad \Gamma \vdash_A A_2 \leq B_2}{\Gamma \vdash_A A_1{\to}A_2 \leq_W B_1{\to}B_2} \quad (\textsc{AWS-Arrow})$$

$$\frac{\Gamma,\, X{\leq}A_1{:}K \vdash_A A_2 \leq B_2 \qquad \mathrm{nf}(A_1) = \mathrm{nf}(B_1) \qquad \mathrm{nf}(K) = \mathrm{nf}(K')}{\Gamma \vdash_A \forall X{\leq}A_1{:}K.A_2 \leq_W \forall X{\leq}B_1{:}K'.B_2} \quad (\textsc{AWS-All})$$

$$\frac{\Gamma,\, X{\leq}A_1{:}K_1 \vdash_A A_2 \leq B_2 \qquad \mathrm{nf}(A_1) = \mathrm{nf}(B_1)}{\Gamma \vdash_A \Lambda X{\leq}A_1{:}K_1.A_2 \leq_W \Lambda X{\leq}B_1{:}K_1'.B_2} \quad (\textsc{AWS-TAbs})$$

Our aim is to prove that this algorithm is sound with respect to the original system on well-formed types: if $\Gamma \vdash_A A \leq B$ and $\Gamma \vdash A, B : K$ then $\Gamma \vdash A \leq B : K$. The problem we encounter is that the original system uses subtyping to check $\Gamma \vdash A, B : K$. Therefore we need algorithmic versions of the judgements involved in proving well-kindedness.

The algorithmic rules for the other judgements are modifications of the $\mathcal{F}^{\omega}_{\leq}$ rules and not of the typed operational semantics. The rules for Context Formation are identical, and in Kind and Type Formation the assumptions of the form $\Gamma \vdash \mathrm{ok}$ are dropped. The rules AK-Π, AT-ALL and AT-TABS need the side condition $X \notin \mathrm{dom}(\Gamma)$ and the rule AT-TAPP uses an arbitrary normalization procedure.

7.3 Algorithmic Context Formation

$$\emptyset \vdash_A \mathrm{ok} \qquad \text{(AC-EMPTY)}$$

$$\frac{\Gamma \vdash_A A : \star \qquad x \notin \mathrm{dom}(\Gamma)}{\Gamma,\, x{:}A \vdash_A \mathrm{ok}} \qquad \text{(AC-VAR)}$$

$$\frac{\Gamma \vdash_A A : K \qquad X \notin \mathrm{dom}(\Gamma)}{\Gamma,\, X{\leq}A{:}K \vdash_A \mathrm{ok}} \qquad \text{(AC-TVAR)}$$

7.4 Algorithmic Kind Formation

$$\Gamma \vdash_A \star \qquad \text{(AK-}\star\text{)}$$

$$\frac{\Gamma,\, X{\leq}A{:}K_1 \vdash_A K_2 \qquad \Gamma \vdash_A A : K_1 \qquad X \notin \mathrm{dom}(\Gamma)}{\Gamma \vdash_A \Pi X{\leq}A{:}K_1.K_2} \qquad \text{(AK-}\Pi\text{)}$$

7.5 Algorithmic Type Formation

$$\Gamma \vdash_A \mathrm{T}_\star : \star \qquad \text{(AT-TOP)}$$

$$\Gamma_1,\, X{\leq}A{:}K,\, \Gamma_2 \vdash_A X : K \qquad \text{(AT-TVAR)}$$

$$\frac{\Gamma \vdash_A A_1 : \star \qquad \Gamma \vdash_A A_2 : \star}{\Gamma \vdash_A A_1 {\rightarrow} A_2 : \star} \qquad \text{(AT-ARROW)}$$

$$\frac{\Gamma,\, X{\leq}A_1{:}K \vdash_A A_2 : \star \qquad \Gamma \vdash_A A_1 : K_1 \qquad X \notin \mathrm{dom}(\Gamma)}{\Gamma \vdash_A \forall X{\leq}A_1{:}K.A_2 : \star} \qquad \text{(AT-ALL)}$$

$$\frac{\Gamma,\, X{\leq}A_1{:}K_1 \vdash_A A_2 : K_2 \qquad \Gamma \vdash_A A : K_1 \qquad X \notin \mathrm{dom}(\Gamma)}{\Gamma \vdash_A \Lambda X{\leq}A_1{:}K_1.A_2 : \Pi X{\leq}A_1{:}K_1.K_2} \qquad \text{(AT-TABS)}$$

$$\frac{\Gamma \vdash_A A : \Pi X{\leq}B{:}K_1.K_2 \qquad \Gamma \vdash_A C \leq B \qquad \Gamma \vdash_A C : K_1' \qquad \mathrm{nf}(K) = \mathrm{nf}(K')}{\Gamma \vdash_A A\,C : K_2[X{\leftarrow}C]} \qquad \text{(AT-TAPP)}$$

7.6 Equivalence of the Algorithm and $\mathcal{F}^\omega_\leq$

PROPOSITION 7.1 (*Correctness of the Algorithm*)

1. If $\Gamma \vdash_A \mathrm{ok}$ then $\Gamma \vdash \mathrm{ok}$.

2. Given $\Gamma \vdash \mathrm{ok}$. If $\Gamma \vdash_A K$ then $\Gamma \vdash K$.

3. Given $\Gamma \vdash \mathrm{ok}$. If $\Gamma \vdash_A A : K$ then $\Gamma \vdash A : K$.

4. Given $\Gamma \vdash A, B : K$. If $\Gamma \vdash_A A \leq_W B$ then $\Gamma \vdash A \leq B : K$.

5. Given $\Gamma \vdash A, B : K$. If $\Gamma \vdash_A A \leq B$ then $\Gamma \vdash A \leq B : K$.

Proof: By induction on derivations, using Soundness (Corollary 5.12), Completeness (Proposition 4.6), and Subject Reduction (Corollary 4.13). □

Proposition 7.2 (*Completeness of the Algorithm*)

1. If $\Gamma \vdash_S$ ok then $\Gamma \vdash_A$ ok.
2. If $\Gamma \vdash_S K$ then $\Gamma \vdash_A K$.
3. If $\Gamma \vdash_S A : K$ then $\Gamma \vdash_A A : K$.
4. If $\Gamma \vdash_S A \leq_W B : K$ then $\Gamma \vdash_A A \leq B$.
5. If $\Gamma \vdash_S A \leq B : K$ then $\Gamma \vdash_A A \leq B$.

Proof: By induction on derivations. The proof is straightforward because all the information in each $\vdash_A$ rules is included in or follows easily from the corresponding $\vdash_S$ rules. □

The last two properties together with the soundness of the semantics (Corollary 5.12) prove that the algorithm is sound and complete with respect to $\mathcal{F}^\omega_\leq$.

Proposition 7.3 (*Equivalence of the Algorithm and* $\mathcal{F}^\omega_\leq$)

1. $\Gamma \vdash_A$ ok iff $\Gamma \vdash$ ok.
2. $\Gamma \vdash$ ok and $\Gamma \vdash_A K$ iff $\Gamma \vdash K$.
3. $\Gamma \vdash$ ok, $\Gamma \vdash_A A : K'$, and $\Gamma \vdash_A K \twoheadrightarrow_{\mathrm{n}} K'$ iff $\Gamma \vdash A : K$.
4. $\Gamma \vdash A, B : K$ and $\Gamma \vdash_A A \leq B$ iff $\Gamma \vdash A \leq B : K$.

Proof: By induction on derivations, using Soundness (Corollary 5.12), Completeness (Proposition 4.6), and Subject Reduction (Corollary 4.13). □

By the equivalence, we can use the following sequence to check whether $\Gamma \vdash A \leq B : K$:

1. check that Γ is a good context, $\Gamma \vdash_A$ ok,
2. infer kinds K' and K'' such that $\Gamma \vdash_A A : K'$ and $\Gamma \vdash_A B : K''$,
3. check that the given kind is well formed, $\Gamma \vdash_A K$,
4. check that K', K'' and the normal form of K (which exists by Strong Normalization (Lemma 4.15)) are syntactically equal,
5. check that $\Gamma \vdash_A A \leq B$.

If any of the steps fails then the statement $\Gamma \vdash A \leq B : K$ is not derivable in $\mathcal{F}^\omega_\leq$, and otherwise it is.

Hence, the only significant result that remains to be proved for $\mathcal{F}^\omega_\leq$ is the decidability of type-checking and subtyping. These follow straightforwardly from the termination of the subtyping algorithm. The details of such a proof should be a simple modification of the proof of decidability of subtyping for $F^\omega_\wedge$ by Compagnoni [17].

8 Related and Future Work

Bruce [6] uses bounded operator abstraction, but does not develop the metatheory. Compagnoni [17] mentions the open problem of studying the metatheory for bounded operator abstraction.

Most type systems with subtyping do not have the circularity between type formation and subtyping mentioned in the introduction: for example, $F^{\omega}_{<:}$ [11, 13, 12, 31, 14], $F^{\omega}_{\wedge}$ [18], and the systems in Abadi and Cardelli's book on objects [2] all separate the two judgements. One system that does have the circularity is $\lambda P_{\leq}$, a system for subtyping with dependent types studied by Aspinall and Compagnoni [4]. There, the authors avoid the interdependency by finding a particular order in which to prove results.

As we mentioned in Section 1, the model construction is based on well-established ideas in dependent type theory. Streicher [34] gives a partial interpretation function to define the categorical semantics of the calculus of constructions, a technique which is now widely used. Coquand and Gallier [20] introduce Kripke-style models to build typed proofs of strong normalization for systems with dependent types. Typed operational semantics has been used to develop the metatheory of UTT, a sophisticated type theory with inductive types, impredicative propositions and type universes [24, 25]. Coquand [19] interprets judgemental equality as a logical relation to show properties of Martin-Löf type theory with $\beta\eta$-equality, similar to our interpretation of the subtyping relation.

It seems to be possible to use the technique developed by the first author for higher-order subtyping [17] for the particular system $\mathcal{F}^{\omega}_{\leq}$ that we study here. We believe that substitution can first be proved simultaneously for the kinding and subtyping judgements for the original system (without the structural rules). This can then be used to prove the subject reduction property for the original system, which in turn is used to establish basic properties of the normal system appropriately formulated for $\mathcal{F}^{\omega}_{\leq}$. However, this approach does not enjoy the advantages of typed operational semantics mentioned in Section 1.2. In particular, the admissibility of the structural rules in Section 2.3 needs to be proved by induction on derivations for each individual rule, the overall proof is delicate and based on a particular order for the results, and the benefits of typed operational semantics for studying properties of reduction such as subject reduction and strong normalization are lost.

There are several directions for future work. The proof here should easily extend to a system with $\beta\eta$-equality, the equality for which typed operational semantics was originally developed. We also believe that the model construction can be extended to cope with Γ-reduction, replacing variables X by their bounds A if $X \leq A : K$ is in Γ, which cannot be done directly in the semantics because of an interdependency of transitivity elimination and context replacement. Furthermore, it seems that the model can deal with a limited form of contravariance for quantification over the kind $\star$ but not over arbitrary kinds. Finally, we have not included recursive types or objects, but Abadi and Cardelli [2] have demonstrated that these do not present difficulties at the level of types, and our proof should extend without any problems.

9 Conclusions

In this paper we have studied $\mathcal{F}^{\omega}_{\leq}$, the first treatment of the metatheory for a system of higher-order subtyping with bounded operator abstraction. We have used techniques for constructing models for dependent type theory to solve problems associated with the weak dependency introduced by bounds in kinds, and we have modeled the subtyping relation directly rather than using a syntactic encoding. We have also used the new tool of typed operational semantics to give simpler proofs for meta-theoretic properties such as substitution, kind correctness, and subject reduction and Church–Rosser for type reduction. Finally, we have shown the equivalence with the algorithmic presentation of the system. Because the techniques introduced are adapted from other contexts and do not involve encodings of syntax, we believe that they are generally applicable.

Acknowledgments

We thank Luca Cardelli for providing encouragement and motivation for this work, and Paul Jackson and Martin Steffen for their comments on drafts of the paper. The first author is funded by a European Community TMR grant, and the second author is funded by a research fellowship from EPSRC.

References

[1] M. Abadi and L. Cardelli. On subtyping and matching. In *ECOOP'95*, pages 145–167. Springer-Verlag, August 1995.

[2] M. Abadi and L. Cardelli. *A Theory of Objects*. Springer-Verlag, 1996.

[3] R. M. Amadio and L. Cardelli. Subtyping recursive types. *ACM Transactions on Programming Languages and Systems*, 15(4):575–631, 1993. A preliminary version appeared in POPL '91 (pp. 104–118), and as DEC Systems Research Center Research Report number 62, August 1990.

[4] D. Aspinall and A. Compagnoni. Subtyping dependent types. In *Eleventh Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA.*, July 27-30 1996. Preliminary version July 1995.

[5] H. P. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *Journal of Symbolic Logic*, 48(4):931–940, 1983.

[6] K. B. Bruce. Typing in object-oriented languages: Achieving expressiveness and safety. Unpublished, June 1996.

[7] K. B. Bruce and G. Longo. A modest model of records, inheritance, and bounded quantification. *Information and Computation*, 87:196–240, 1990. Also in [27]. An earlier version appeared in the proceedings of the IEEE Symposium on Logic in Computer Science, 1988.

[8] K. B. Bruce and J. Mitchell. PER models of subtyping, recursive types and higher-order polymorphism. In *Proceedings of the Nineteenth ACM Symposium on Principles of Programming Languages*, Albequerque, NM, January 1992.

[9] K. B. Bruce, A. Schuett, and R. van Gent. Polytoil: a type-safe polymorphic object-oriented language. In *ECOOP'95*. Springer-Verlag, August 1995.

[10] L. Cardelli. A semantics of multiple inheritance. *Information and Computation*, 76:138–164, 1988. Preliminary version in *Semantics of Data Types*, Kahn, MacQueen, and Plotkin, eds., Springer-Verlag LNCS 173, 1984.

[11] L. Cardelli. Types for data-oriented languages. In *First Conference on Extending Database Technology*, volume 303 of *Lecture Notes in Computer Science*. Springer-Verlag, May 1988.

[12] L. Cardelli. Notes about $\mathrm{F}^{\omega}_{<:}$. Unpublished manuscript, October 1990.

[13] L. Cardelli. Typeful programming. In E. J. Neuhold and M. Paul, editors, *Formal Description of Programming Concepts*. Springer-Verlag, 1991. An earlier version appeared as DEC Systems Research Center Research Report #45, February 1989.

[14] L. Cardelli and G. Longo. A semantic basis for Quest. *Journal of Functional Programming*, 1(4):417–458, October 1991. Preliminary version in ACM Conference on Lisp and Functional Programming, June 1990. Also available as DEC SRC Research Report 55, Feb. 1990.

[15] L. Cardelli and P. Wegner. On understanding types, data abstraction, and polymorphism. *Computing Surveys*, 17(4), December 1985.

[16] A. B. Compagnoni. Decidability of higher-order subtyping with intersection types. In *Proceedings of the Annual Conference of the European Association for Computer Science Logic, CSL'94, Kazimierz, Poland*, number 933 in Lecture Notes in Computer Science. Springer-Verlag, June 1995. Preliminary version available as University of Edinburgh technical report ECS-LFCS-94-281, January 1994, under the title "Subtyping in $F^{\omega}_{\wedge}$ is decidable".

[17] A. B. Compagnoni. *Higher-Order Subtyping with Intersection Types*. PhD thesis, University of Nijmegen, The Netherlands, January 1995. ISBN 90-9007860-6.

[18] A. B. Compagnoni and B. C. Pierce. Higher-order intersection types and multiple inheritance. *Mathematical Structures in Computer Science*, 6:469–501, 1996. Preliminary version available under the title *Multiple Inheritance via Intersection Types* as University of Edinburgh technical report ECS-LFCS-93-275 and Catholic University Nijmegen computer science technical report 93-18, Aug. 1993.

[19] T. Coquand. An algorithm for testing conversion in type theory. In G. Huet and G. Plotkin, editors, *Logical Frameworks*. Cambridge University Press, 1991.

[20] T. Coquand and J. Gallier. A proof of strong normalization for the theory of constructions using a Kripke-like interpretation. In *Workshop on Logical Frameworks–Preliminary Proceedings*, 1990.

[21] P.-L. Curien and G. Ghelli. Coherence of subsumption: Minimum typing and type-checking in $\mathrm{F}_{\leq}$. *Mathematical Structures in Computer Science*, 2:55–91, 1992.

[22] M. Dezani-Ciancaglini and I. Margaria. A characterisation of F-complete type assignments. *Theoretical Computer Science*, 45:121–157, 1986.

[23] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge University Press, 1989.

[24] H. Goguen. *A Typed Operational Semantics for Type Theory.* PhD thesis, University of Edinburgh, Aug. 1994.

[25] H. Goguen. The metatheory of *UTT*. In *Types for Proofs and Programs*, volume 996 of *Lecture Notes in Computer Science*, pages 60–82, Baastad, Sweden, June 1995. Springer–Verlag.

[26] H. Goguen. Typed operational semantics. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 186–200. Springer–Verlag, 1995.

[27] C. A. Gunter and J. C. Mitchell. *Theoretical Aspects of Object-Oriented Programming: Types, Semantics, and Language Design.* The MIT Press, 1994.

[28] P. Martin-Löf. An intuitionistic theory of types, 1972. Unpublished manuscript.

[29] J. McKinna and R. Pollack. Pure type systems formalized. In M. Bezem and J. F. Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, pages 289–305. Springer–Verlag, LNCS 664, Mar. 1993.

[30] J. Mitchell. Type systems for programming languages. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science.* North Holland, 1990.

[31] J. C. Mitchell. Toward a typed foundation for method specialization and inheritance. In *Proceedings of the 17th ACM Symposium on Principles of Programming Languages*, pages 109–124, January 1990. Also in [27].

[32] B. C. Pierce. *Programming with Intersection Types and Bounded Polymorphism.* PhD thesis, Carnegie Mellon University, December 1991. Available as School of Computer Science technical report CMU-CS-91-205.

[33] M. Steffen and B. Pierce. Higher-order subtyping. In *IFIP Working Conference on Programming Concepts, Methods and Calculi (PROCOMET)*, June 1994. An earlier version appeared as University of Edinburgh technical report ECS-LFCS-94-280 and Universität Erlangen-Nürnberg Interner Bericht IMMD7-01/94, February 1994.

[34] T. Streicher. *Semantics of Type Theory: Correctness, Completeness and Independence Results.* Birkhäuser, 1991.

[35] M. Takahashi. Parallel reductions in λ-calculus. *Information and Computation*, 118:120–127, 1995.

[36] A. K. Wright and M. Felleisen. A syntactic approach to type soundness. *Information and Computation*, 115(1):38–94, 15 nov 1994.