MAX-PLANCK-INSTITUT FÜR INFORMATIK

Structural Decidable Extensions of Bounded Quantification

Sergei Vorobyov

MPI–I–94–257 October 1994

INFORMATIK

Im Stadtwald
D 66123 Saarbrücken
Germany

Author's Address

Sergei Vorobyov (sv@mpi-sb.mpg.de),
Max–Planck–Institut für Informatik
Im Stadtwald
D-66123 Saarbrücken
Germany
(http://www.mpi-sb.mpg.de/guide/staff/sv/sv.html)

Publication Notes

To appear in the Proceedings of the 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'95), January 23-25, 1995, San Francisco, U.S.A. Preliminary versions appeared as Technical Report CRIN–94–R-121, July 1994 (*Centre de Recherche en Informatique de Nancy*, France) and Research Report INRIA–94–RR–2309, September 1994 *Institut National de Recherche en Informatique et en Automatique*, France.

Acknowledgements

I am greatly indebted to Luca Cardelli, Benjamin Pierce, Martín Abadi, Roberto Amadio, Philippe de Groote, Didier Galmiche, Jean-Luc Rémy, Hubert Comon, Michel Parigot for invaluable remarks, ideas, and discussions. I am grateful to my anonymous referee for his substantial help. To produce proof trees I used Paul Taylor's LaTeXmacro package. This work was mainly done when I was at CRIN (Centre National de Recherche en Informatique de Nancy, France), which provided me the excellent research opportunities.

Abstract

We show how the subtype relation of the well-known system $F_{\leq}$, the second-order polymorphic λ-calculus with bounded universal type quantification and subtyping, due to Cardelli, Wegner, Bruce, Longo, Curien, Ghelli, proved undecidable by Pierce (POPL'92), can be interpreted in the (weak) monadic second-order theory of one (Büchi), two (Rabin), several, or infinitely many successor functions. These **(W)SnS**-interpretations show that the undecidable system Fsub possesses consistent decidable extensions, i.e., Fsub is not essentially undecidable (Tarski, 1949).

We demonstrate an infinite class of structural decidable extensions of $F_{\leq}$, which combine traditional subtype inference rules with the above **(W)SnS**-interpretations. All these extensions, which we call systems $F_{\leq}^{SnS}$, are still more powerful than $F_{\leq}$, but less coarse than the direct **(W)SnS**-interpretations.

The main distinctive features of the systems $F_{\leq}^{SnS}$ are: 1) decidability, 2) closure w.r.t. transitivity; 3) structuredness, e.g., they never subtype a functional type to a universal one or vice versa, 4) they all contain the powerful rule for subtyping boundedly quantified types.

Structural Decidable Extensions of Bounded Quantification

Sergei G. Vorobyov *

Max-Planck-Institut für Informatik

Im Stadtwald, D-66123, Saarbrücken, Germany

(e-mail: sv@mpi-sb.mpg.de)

Abstract

We show how the subtype relation of the well-known system $F_{\leq}$, the second-order polymorphic λ-calculus with bounded universal type quantification and subtyping, due to Cardelli, Wegner, Bruce, Longo, Curien, Ghelli [6, 2, 8], proved undecidable by Pierce [12], can be interpreted in the (weak) monadic second-order theory of one (Büchi), two (Rabin), several, or infinitely many successor functions [13, 14]. These **(W)SnS**-interpretations show that the undecidable system $F_{\leq}$ possesses consistent decidable extensions, i.e., $F_{\leq}$ *is not essentially undecidable* (Tarski *et. al.*, 1949, [17]).

We demonstrate an infinite class of "*structural*" decidable extensions of $F_{\leq}$, which combine traditional subtype inference rules with the above **(W)SnS**-interpretations. All these extensions, which we call systems $F_{\leq}^{SnS}$, are still more powerful than $F_{\leq}$, but less coarse than the direct **(W)SnS**-interpretations:

$$F_{\leq} \subset F_{\leq}^{SnS} \subset (\mathbf{W})\mathbf{SnS}\text{-}interpretations$$

The main distinctive features of the systems $F_{\leq}^{SnS}$ are:
1) decidability, 2) closure w.r.t. transitivity; 3) structuredness, e.g., they never subtype a functional type to a universal one or vice versa, 4) they all contain the powerful rule for subtyping boundedly quantified types:

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \qquad \Gamma, \alpha \leq \tau_1 \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash (\forall\alpha \leq \sigma_1 \,.\, \sigma_2) \leq (\forall\alpha \leq \tau_1 \,.\, \tau_2)} \quad (All)$$

Key words: second-order polymorphic typed λ-calculus, subtyping, system $F_{\leq}$, bounded universal type quantification, parametric and inheritance polymorphisms, (un-)decidability, essential undecidability, (weak) monadic second-order theory of several successor functions **(W)SnS**.

1 Introduction

The advantages and usefulness of strict typing disciplines in programming with static typing and rigid compile-time type control have been widely accepted, studied, and advocated in Software Engineering [10, 6, 4, 11] since creation of Simula-67, Algol-68, Pascal, Clu, Alphard, Modula, ML, Ada, etc. Typeful programming should be based on powerful and, preferably, decidable type systems.

The system $F_{\leq}$ is the polymorphic second-order typed λ-calculus with subtyping, combining the universal (or parametric) polymorphism of Girard's system F with Cardelli's calculus of subtyping (inheritance polymorphism [3]). Introduced in [6], later improved, simplified, and investigated by many researchers [2, 1, 8, 12, 7, 5], the system $F_{\leq}$ serves a core calculus of type systems with subtyping and a model to represent polymorphic and object-oriented features in programming languages.

$F_{\leq}$ is an extension of F with subtyping. In addition to the usual functional and universal type formation of F, the system $F_{\leq}$ allows one to form boundedly quantified types: $\forall\alpha \leq bound \,.\, body$. Such type is a function on types transforming any subtype σ of a *bound* into a type $body[\sigma/\alpha]$. As $F_{\leq}$ also contains the largest type $\top$, the unbounded type quantification of F is included as a particular case: $\forall\alpha \leq \top \,.\, \sigma$.

The system $F_{\leq}$ consists of two components. The first one axiomatizes the subtyping relation on types $\Gamma \vdash \sigma \leq \tau$. The second generates the typing relation $\Gamma \vdash t : \sigma$. Both components interact by means of the rules as (*Subsumption*), allowing one to derive the judgment $\Gamma \vdash t : \tau$ from $\Gamma \vdash t : \sigma$ and $\Gamma \vdash \sigma \leq \tau$.

In [12] Pierce proved that already the subtyping component of $F_{\leq}$ is undecidable, and hence the typing relation in $F_{\leq}$ is undecidable too. Using Ghelli's example of divergence of $F_{\leq}$-subtyping algorithm (mainly due to the subtle interaction between the quantifier rule (*All*) above and transitivity), he succeeded to encode instances of the termination problem into $F_{\leq}$-subtyping judgments.

Given an undecidable theory T one usually tries to *weaken* it to get a decidable *subtheory* $T_{dec} \subseteq T$. Accordingly, attempts were made to restrict $F_{\leq}$ to get decidable subsystems. In [7] the general quantifier rule (*All*) above was replaced by its weaker version:

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \quad \Gamma, \alpha \leq \top \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash (\forall\alpha \leq \sigma_1 \,.\, \sigma_2) \leq (\forall\alpha \leq \tau_1 \,.\, \tau_2)} \quad (All\text{-}Top)$$

Subtyping in the resulting subsystem $F_{\leq}^{\top} \subset F_{\leq}$ is decidable. In [9] a decidable subsystem of $F_{\leq}$ is obtained by

*This work was done while the author was at the *Centre de Recherche en Informatique de Nancy, France* restricting bounds in bounded quantification to be $\top$-free (with some relaxations to allow unbounded quantification). An extensive discussion of different other weakenings of the powerful rule (*All*) is contained in [7].

For an undecidable theory T there sometimes exists another possibility, to *reinforce* it (instead of *weakening*) in order to obtain a *consistent decidable extension* $T_{dec} \supseteq T$. This works only if T is *not essentially undecidable*, i.e., possesses consistent decidable extensions (A. Tarski, 1949, [17]).

Curiously enough, $F_{\leq}$ appears to be undecidable, but *not essentially* [18], with infinitely many nontrivial consistent decidable extensions. This reopens the possibility for obtaining good decidable systems relative to $F_{\leq}$ *without sacrificing* the general quantifier rule (*All*) or somehow *restricting* the form of bounds in bounded quantification.

The first infinite class of such extensions was introduced in [18], where it was shown that there exist infinitely many ways to translate $F_{\leq}$-subtyping judgments into formulas of Rabin's **S2S**. Each such translation maps the $F_{\leq}$-axioms to valid **S2S**-formulas, and each $F_{\leq}$-inference rule preserves validity with respect to any **S2S**-translation. It follows that everything provable in $F_{\leq}$ is valid in any **S2S**-interpretation. Consequently, $F_{\leq}$ is not essentially undecidable; any **S2S**-translation is a consistent decidable extension of $F_{\leq}$. **S2S**-interpretations generalize for recursive types [19].

Precautions, however, should be taken concerning consistency. For theories based on predicate calculus *consistent* means "*do not prove everything*". For theories, which are not based on predicate calculus, like $F_{\leq}$, *consistent* might mean "*do not subtype any pair of types*" (weak consistency) or "*do not subtype too many types*" (strong consistency).

S2S-interpretations appeared to be weakly, but not strongly consistent. They are *coarse* in the sense that they do not make fine distinction between differently structured types, and subtype too many of them, which is undesirable in strict typing disciplines. In this paper we remedy this drawback by combining our **S2S**-interpretations with the traditional $F_{\leq}$-like subtype inference rules. These rules guarantee the so-called "strict structural subtyping", where the subtype relation is defined by co(ntra)variant induction on type structure. This prevents us from subtyping differently structured types, e.g., universal and functional ones.

The main idea of our systems $F_{\leq}^{SnS}$ is that they disable the infinite alternations of applications of the rule (*All*) and the transitivity rule. This alternation is the source of non-termination and undecidability of $F_{\leq}$, [12]. Instead, we prune proof tree branches, which may lead to infinite alternations, and decide the remaining judgments by interpreting them in **(W)SnS**. Of course, as $F_{\leq}$ is undecidable, and $F_{\leq}^{SnS}$ are decidable extensions of $F_{\leq}$, sometimes they accept $F_{\leq}$-unprovable judgments. But this is a reasonable price for attaining decidability.

The scenario of the presentation is the following. Section 2 recalls the system $F_{\leq}$. (Un)decidability results concerning $F_{\leq}$ are listed in Section 3. Section 4 introduces systems $F_{\leq}^{SnS}$. Section 5 describes the decision procedure. In Section 6 we show infinitely many ways to interpret the subtype relation in any **(W)SnS**. Section 7 discusses the consistency of $F_{\leq}^{SnS}$. In Section 8 we explain the rule inversion principle, the main tool of our proofs of the inclusion $F_{\leq} \subset F_{\leq}^{SnS}$ and the transitivity of $F_{\leq}^{SnS}$. In Sections 9 and 10 we show that the inversion principle does not hold for **SnS**-interpretations, but holds for systems $F_{\leq}^{SnS}$. In Sections 11, 12, and 13 we prove the inclusions $F_{\leq} \subset F_{\leq}^{SnS} \subset$ **(W)SnS**-*interpretations* and the transitivity of all $F_{\leq}^{SnS}$. Section 14 discusses further improvements of $F_{\leq}^{SnS}$. In Section 15 we sketch problems for future research. Appendices A and B contain the reference material on second-order monadic theories and on Curien-Ghelli's algorithmic variant of $F_{\leq}$. The proofs are collected in Appendix C.

In this paper we deal only with the subtyping relation. Combinations with typing and related problems, like subject reduction [20], typing proof normalization, the least type property, strong normalization are in the course of study and will be considered elsewhere.

Added in Proof. In [21] we continued the study of decidable extensions of the $F_{\leq}$ subtyping relation and developed the general theory of converging hierarchies of structural decidable extensions of the $F_{\leq}$-subtyping. The systems $F_{\leq}^{SnS}$ presented in this paper form just the first level of the hierarchies from [21]. In [22] we combined these hierarchies with the standard $F_{\leq}$ term typing rules and obtained an infinite family of the extensions of the polymorphic system $F_{\leq}$ where *both subtyping and typing are decidable.*

2 System $F_{\leq}$

For complete and exact reference see, e.g., [8, 12, 5]. We just briefly remind the essential definitions, retaining the notation of [12].

Definition 2.1 (Types) *The set of* $F_{\leq}$-types *is defined by the following abstract grammar:*

$$\mathbb{T} \;\equiv_{df}\; \mathbb{V} \mid \top \mid \mathbb{T} \rightarrow \mathbb{T} \mid \forall \mathbb{V} \leq \mathbb{T} \,.\, \mathbb{T}$$

where:

1. $\mathbb{V}$ *is a set of* type variables *denoted by Greek letters* α, β, γ;
2. $\top$ *is the largest type majorazing any other type,* $\sigma \leq \top$;
3. $\rightarrow$ *is the functional type constructor,* $\sigma \rightarrow \tau$ *is the type of functions with domain of type* σ *and codomain of type* τ;
4. $\forall \alpha \leq \rho \,.\, \tau$ *is a polymorphic* boundedly quantified *type, i.e., a function assigning to each subtype* σ *of* ρ, $\sigma \leq \rho$, *the type* $\tau[\sigma/\alpha]$ *obtained from* τ *by substituting* σ *instead of free occurrences of* α *(with usual non-clashing preconditions on free variables). In* $\forall \alpha \leq \rho \,.\, \tau$ *the bound* ρ *does not contain* α *free.*

The letters τ, σ, ρ *from the end of the Greek alphabet denote arbitrary (variable or compound)* $F_{\leq}$*-types;* $\forall \beta \,.\, \tau$ *abbreviates* $\forall \beta \leq \top . \tau$; $FV(\sigma)$ *denotes the set of free variables in* σ.

Definition 2.2 (Contexts) *An* $F_{\leq}$-context *is an ordered sequence* $\alpha_1 \leq \sigma_1, \ldots, \alpha_n \leq \sigma_n$ *of* $\leq$*-relations between type variables and* $F_{\leq}$*-types such that:*

1. *all* α_i *are different type variables, and*

2. for each i, $FV(\sigma_i) \subseteq \{\alpha_1, \ldots, \alpha_{i-1}\}$.

Contexts are denoted by capital Greek Γ. $Dom(\Gamma)$ is the set of type variables appearing to the left of $\leq$ in Γ. We write $\Gamma(\alpha) = \sigma$ if Γ contains $\alpha \leq \sigma$ and call σ a bound *of α in Γ. We define $\Gamma^*(\alpha)$ as $\Gamma(\alpha)$ if the latter is not a variable, and as $\Gamma^*(\Gamma(\alpha))$ otherwise.* □

Definition 2.3 (Subtyping Judgments)
*An $F_{\leq}$-*subtyping judgment *is a figure of the form:*

$$\Gamma \vdash \sigma \leq \tau,$$

where $FV(\sigma) \cup FV(\tau) \subseteq Dom(\Gamma)$. □

The intuitive semantics of a judgment $\Gamma \vdash \sigma \leq \tau$ is: σ is a subtype of τ provided that all α_i mentioned in Γ are subtypes of their respective bounds σ_i.

Definition 2.4 (Subtyping Rules) *The $F_{\leq}$-subtyping relation is generated by the system of 3 axioms and 3 inference rules, shown in Figure 1.*

$$\Gamma \vdash \tau \leq \tau \qquad (Refl)$$

$$\Gamma \vdash \tau \leq \top \qquad (Top)$$

$$\Gamma \vdash \alpha \leq \Gamma(\alpha) \qquad (TVar)$$

$$\frac{\Gamma \vdash \tau_1 \leq \tau_2 \quad \Gamma \vdash \tau_2 \leq \tau_3}{\Gamma \vdash \tau_1 \leq \tau_3} \qquad (Trans)$$

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \quad \Gamma \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash \sigma_1 \to \sigma_2 \leq \tau_1 \to \tau_2} \qquad (Arrow)$$

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \quad \Gamma, \alpha \leq \tau_1 \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash (\forall \alpha \leq \sigma_1 . \sigma_2) \leq (\forall \alpha \leq \tau_1 . \tau_2)} \qquad (All)$$

Figure 1: $F_{\leq}$ subtyping axioms and inference rules

Let $\vdash_{F_{\leq}}$ denote the least three-place relation $\Gamma \vdash \sigma \leq \tau$ containing all particular cases of the $F_{\leq}$-axioms and closed with respect to the $F_{\leq}$-inference rules. Sometimes, by abusing notation, we denote by $F_{\leq}$ the set of subtyping judgments provable in $F_{\leq}$. □

Definition 2.5 (Variants of $F_{\leq}$) *1) Original Fun [6] replaces the (All) rule by the weaker rule (All-Fun) (Figure 2).*

2) System $F_{\leq}^{\top}$ [7] replaces the rule (All) by its particular case (All-Top) (Figure 2).

3) System $F_{\leq}{}^{local}$ [7] replaces the rule (All) by its modification (All-local) (Figure 2).

By $\vdash_{Fun}$, $\vdash_{F_{\leq}^{\top}}$ and $\vdash_{F_{\leq}{}^{local}}$ we denote the corresponding subtyping relations. □

$$\frac{\Gamma, \alpha \leq \rho \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash (\forall \alpha \leq \rho . \sigma_2) \leq (\forall \alpha \leq \rho . \tau_2)} \qquad (All\text{-}Fun)$$

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \quad \Gamma, \alpha \leq \top \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash (\forall \alpha \leq \sigma_1 . \sigma_2) \leq (\forall \alpha \leq \tau_1 . \tau_2)} \qquad (All\text{-}Top)$$

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \quad \Gamma, \alpha \leq \sigma_1 \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash (\forall \alpha \leq \sigma_1 . \sigma_2) \leq (\forall \alpha \leq \tau_1 . \tau_2)} \qquad (All\text{-}local)$$

Figure 2: Variants of the (*All*) rule

3 (Un)Decidability

The interesting facts about $F_{\leq}$ are:

Theorem 3.1 (Undecidability of $F_{\leq}$, [12]) *The relation $\vdash_{F_{\leq}}$ is undecidable.* □

The weakenings of $F_{\leq}$ are however decidable:

Theorem 3.2 (Decidability of Fun and $F_{\leq}^{\top}$ [7]) *Both relations $\vdash_{Fun}$ and $\vdash_{F_{\leq}^{\top}}$ are decidable.* □

Nothing is known about decidability of $F_{\leq}{}^{local}$ [7].

In [18] we demonstrated that the decidability of $F_{\leq}$ could be reached also by *reinforcement*, and not only by *weakening*, as opposed to systems $F_{\leq}^{\top}$ and Fun.

Definition 3.3 (Essential Undecidability, [17]) *A consistent theory T is* essentially undecidable *iff it has no consistent decidable extensions $T' \supseteq T$.* □

Definition 3.4 (Consistency) *An extension of $F_{\leq}$ is* consistent *iff it is closed with respect to the $F_{\leq}$ inference rules and does not subtype any two types.* □

Remarks. 1) Further we replace "any two types" by "any two differently structured types" getting the stronger consistency. 2) As we are interested only in the extensions of $F_{\leq}$, the closure with respect to the $F_{\leq}$-inference rules seems natural and meaningful. It would not be the case for $F_{\leq}^{\top}$ and Fun. □

Theorem 3.5 ($F_{\leq}$ Is Not Essentially Undecidable, [18]) *There exist infinitely many different consistent decidable extensions of $\vdash_{F_{\leq}}$.* □

This result was obtained by interpreting the $F_{\leq}$-subtyping relation in **S2S**, the monadic second-order logic of two successors due to M. Rabin [13, 14]. The corresponding infinite class of extensions of $F_{\leq}$ (which we call the **S2S**-interpretations) and their properties are studied in [18].

The main objection (by L. Cardelli and others) against these extensions was that they were too coarse and non-structural. **S2S**-interpretations subtype too many types, sometimes differently structured ones (i.e., universal and functional ones).

In this paper we introduce a new infinite class of decidable extensions of $F_{\leq}$ refining the **S2S**-interpretations . We call these extensions *systems* $F_{\leq}^{SnS}$. We also (re)introduce the **S2S**-interpretations in a slightly more general setting and call them **SnS**-interpretations (with **S2S** being a particular case of **SnS** for $n = 2$). We prove that all systems $F_{\leq}^{SnS}$ are *more powerful* than $F_{\leq}$, but being structural (they do not subtype differently structured types any more), they are less coarse than **SnS**-interpretations:

$$F_{\leq} \subset F_{\leq}^{SnS} \subset \mathbf{SnS}\text{-}interpretations$$

Again note that the decidable system $F_{\leq}^{\top}$ introduced in [7] is *weaker* than $F_{\leq}$: $F_{\leq}^{\top} \subset F_{\leq}$.

4 System $F_{\leq}^{SnS}$

Definition 4.1 *The system $F_{\leq}^{SnS}$ is defined by the collection of subtyping axioms and inference rules shown in Figure 3, supposed to be applied* bottom-up *in the* order *of their presentation.*

$$***\ See\ Figure\ 3\ ***$$

The $DECIDE$ component in the rule (Var-All-$Decide$) and the whole $F_{\leq}^{SnS}$-decision procedure are described in the following Sections. □

Roughly speaking, the system $F_{\leq}^{SnS}$ is $F_{\leq}$ without the general transitivity rule ($Trans$) replaced by a built-in decision procedure $DECIDE$.

Remarks and Explanations

1. Our intention is to define the *decision* and not semidecision procedure for subtyping judgments. That is why we are going to apply rules bottom-up and introduce two constants $TRUE$ and $FALSE$ to treat both the accepting and rejecting cases.

2. Rules ($Refl$), (Top), and ($TVar$) correspond exactly to their $F_{\leq}$ counterparts. We formulate them as rules with the premises $TRUE$ just to be able to treat symmetrically the negative case $FALSE$ in other rules of $F_{\leq}^{SnS}$.

3. Rules ($Arrow$) and (All) are the same as in $F_{\leq}$.

4. Motivation for the rules (Top-L) and ($TVar$-R-2) is: the conclusions of these rules are NOT *provable* in $F_{\leq}$ (Proposition 4.2).

5. Motivation for the rules ($\forall \not\leq \rightarrow$) and ($\rightarrow \not\leq \forall$) is the same: the conclusions of these rules are underivable in $F_{\leq}$.

6. The (Var-$Arrow$) rule is just a half (with only arrow-types on the right of $\leq$) of Curien-Ghelli's algorithmic transitivity rule ($AlgTrans$), see [8] and Appendix B.

7. The crucial difference with $F_{\leq}$ is the *absence of the general rule* ($Trans$) or of its algorithmic equivalent ($AlgTrans$) for universal types (see the rule (Var-All) below). Transitivity in this case is dealt separately, by means of a $DECIDE$ procedure. Note that we do not weaken the general $F_{\leq}$ quantifier rule (All), which remains the same as in $F_{\leq}^{SnS}$.

8. The built-in procedure $DECIDE$ appearing in the premise of the rule (Var-All-$Decide$) is a parameter of the system. Below we define infinitely many different such procedures. Note, in particular, that if we define the $DECIDE$ procedure recursively, as $F_{\leq}^{SnS}$ plus the second half of Curien-Ghelli's transitivity rule:

$$\frac{\Gamma \vdash \Gamma(\alpha) \leq (\forall\beta \leq \sigma\,.\,\tau)}{\Gamma \vdash \alpha \leq (\forall\beta \leq \sigma\,.\,\tau)} \qquad (Var\text{-}All)$$

then we will get *exactly* $F_{\leq}$! □

Proposition 4.2 *Subtyping judgments of the forms:*

1. $\Gamma \vdash \top \leq \tau \quad (\tau \neq \top)$,
2. $\Gamma \vdash \sigma \leq \alpha \quad$ (σ *non-variable*, α *variable*),

where Γ *is any context, are not provable in* $F_{\leq}$. □

Proof . See Appendix C.1. □

5 Decision Procedure

The rules of the system $F_{\leq}^{SnS}$ read bottom-up can be seen as a decision procedure (with a built-in $DECIDE$ oracle). Given a subtyping judgment, the rules of $F_{\leq}^{SnS}$ apply deterministically in ordered manner (e.g., (Var-All-$Decide$) does not apply before (Var-All-2)). The rule application process always terminates, provided that the built-in $DECIDE$ procedure is finitely terminating, and this is the fundamental difference with $F_{\leq}$, see [12].

Proposition 5.1 (Finite Termination of $F_{\leq}^{SnS}$) *For every subtyping judgment $\Gamma \vdash \sigma \leq \tau$ any $F_{\leq}^{SnS}$-proof tree is finite.* □

Proof . The complexity of judgments decreases as one moves bottom-up. □

So the termination of the whole decision procedure depends on termination of its $DECIDE$ component.

Irreducible leaves of $F_{\leq}^{SnS}$-proof trees are either:

1. $TRUE$ or
2. $FALSE$ or
3. of the form $DECIDE(\ J\ )$, where J is a subtyping judgment in the $F_{\leq}^{SnS}$-*normal form*, i.e.:

$$J \equiv_{df} \alpha_1 \leq \sigma_1 \ldots \alpha_n \leq \sigma_n \vdash \beta \leq \tau, \qquad (3)$$

where $\alpha_1, \ldots, \alpha_n, \beta$ are type variables, $\sigma_1, \ldots, \sigma_n$ are arbitrary types, and τ is a universal type.

$$\frac{TRUE}{\Gamma \vdash \sigma \leq \sigma} \qquad (Refl)$$

$$\frac{TRUE}{\Gamma \vdash \sigma \leq \top} \qquad (Top)$$

$$\frac{FALSE}{\Gamma \vdash \top \leq \tau} \ (for\ \tau \neq \top) \qquad (Top\text{-}L)$$

$$\frac{\Gamma \vdash \Gamma(\beta) \leq \alpha}{\Gamma \vdash \beta \leq \alpha} \ (for\ different\ variables\ \alpha,\ \beta) \qquad (TVar\text{-}R\text{-}1)$$

$$\frac{FALSE}{\Gamma \vdash \sigma \leq \alpha} \ (\sigma\ non\text{-}variable,\ \alpha\ variable) \qquad (TVar\text{-}R\text{-}2)$$

$$\frac{FALSE}{\Gamma \vdash (\forall \alpha \leq \sigma_1 \,.\, \sigma_2) \leq (\tau_1 \to \tau_2)} \qquad (\forall \not\leq \to)$$

$$\frac{FALSE}{\Gamma \vdash (\sigma_1 \to \sigma_2) \leq (\forall \alpha \leq \tau_1 \,.\, \tau_2)} \qquad (\to \not\leq \forall)$$

$$\frac{\Gamma \vdash \Gamma(\alpha) \leq \sigma \to \tau}{\Gamma \vdash \alpha \leq \sigma \to \tau} \qquad (Var\text{-}Arrow)$$

$$\frac{TRUE}{\Gamma \vdash \alpha \leq \Gamma(\alpha)} \qquad (TVar)$$

$$\frac{\Gamma \vdash \Gamma(\alpha) \leq (\forall \beta \leq \sigma \,.\, \tau)}{\Gamma \vdash \alpha \leq (\forall \beta \leq \sigma \,.\, \tau)} \ (if\ \Gamma(\alpha)\ is\ a\ variable) \qquad (Var\text{-}All\text{-}1)$$

$$\frac{\Gamma \vdash FALSE}{\Gamma \vdash \alpha \leq (\forall \beta \leq \sigma \,.\, \tau)} \ (if\ \Gamma(\alpha)\ is\ \top\ or\ an\ \to\text{-}type) \qquad (Var\text{-}All\text{-}2)$$

$$\frac{DECIDE(\,\Gamma \vdash \alpha \leq (\forall \beta \leq \sigma \,.\, \tau)\,)}{\Gamma \vdash \alpha \leq (\forall \beta \leq \sigma \,.\, \tau)} \qquad (Var\text{-}All\text{-}Decide)$$

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \qquad \Gamma \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash \sigma_1 \to \sigma_2 \leq \tau_1 \to \tau_2} \qquad (Arrow)$$

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \qquad \Gamma, \alpha \leq \tau_1 \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash (\forall \alpha \leq \sigma_1 \,.\, \sigma_2) \leq (\forall \alpha \leq \tau_1 \,.\, \tau_2)} \qquad (All)$$

Figure 3: System $F_{\leq}^{SnS}$

Obviously:

- if all leaves of a $F_{\leq}^{SnS}$-proof tree are $TRUE$, we declare the input judgment valid;
- if one of the leaves of $F_{\leq}^{SnS}$-proof tree is $FALSE$, we declare the input judgment invalid;
- otherwise, before announcing our verdict we analyze $F_{\leq}^{SnS}$-normal forms (3) using the built-in $DECIDE$ procedure.

To decide normal forms (3) we use a method [18] of interpretations in monadic second-order theories of successor functions [14]:

- first, we compile $F_{\leq}^{SnS}$-normal forms (3) in a monadic second-order theory,
- second, we decide them using a decision procedure for this theory.

Therefore, instead of remaining in the undecidable $F_{\leq}$ we forget it and work in the decidable $F_{\leq}^{SnS}$, which replaces the transitivity rule ($Trans$) by the transitivity implicitly present in a monadic second-order theory. As we show below, the proper choices of the $DECIDE$ component lead to *decidable extensions* of $F_{\leq}$ (Theorem 11.1), *closed with respect to transitivity* (Theorem 12.1).

6 Interpreting $F_{\leq}^{SnS}$-Normal Forms in SnS

In [18] we introduced an infinite class of direct interpretations of $F_{\leq}$ into **S2S**, the monadic second-order arithmetic of two successor functions [13, 14]. These direct **S2S**-interpretations *do not use any inference rules* (as opposed to $F_{\leq}$ or $F_{\leq}^{SnS}$), immediately translating $F_{\leq}$-judgments into **S2S**-formulas. Like this we established that $F_{\leq}$ possesses infinitely many different consistent decidable extensions, i.e., is not essentially undecidable.

The drawback of the direct **S2S**-interpretations of $F_{\leq}$ is that they subtype too many types (see [18] and below), in particular, differently structured types. The systems $F_{\leq}^{SnS}$ are more subtle. By their very definition they do not subtype differently structured types. They cannot prove a subtyping between, say, an $\rightarrow$-type and a $\forall$-type. The systems $F_{\leq}^{SnS}$ apply the method of interpretations *only to normal forms*, i.e., to judgments of the form (3) inside the $DECIDE$ procedure.

There is only a minor difference in defining the **S2S**-interpretations only for normal forms (3) and for general $F_{\leq}$-subtyping judgments, so we give a complete definition of **S2S**-interpretations of $F_{\leq}$. Also, **S2S**-interpretations generalize straightforwardly to **SnS**-interpretations for arbitrary $n \in \mathbb{N}$ or even **S**ω**S**.

Choose and fix *any monadic second-order theory of successor function(s)*, say, Büchi arithmetic **S1S**, Rabin's arithmetic **S2S**, ..., **SnS**, **S**ω**S**, or their weak counterparts, with second-order quantifications restricted to finite sets (see Appendix A).

The intuition behind interpretations of $F_{\leq}$ into **SnS** is extremely simple. We interpet the $F_{\leq}$ types *as propositions* of **SnS**. Each $F_{\leq}$-type σ is assigned a **SnS**-formula $S(x)$ with just one free object variable x, and each subtyping relation $\sigma \leq \tau$ is translated into $\forall x(S(x) \supset T(x))$, where $S(x)$ and $T(x)$ are **SnS**-formulas assigned to types σ and τ.

Our translation satisfies the following properties:

1. all axioms of $F_{\leq}$ are transformed into valid formulas of **SnS**;
2. all $F_{\leq}$-inference rules preserve validity with respect to any **SnS**, i.e., whenever both premises of a rule are translated into valid **SnS**-formulas, then the conclusion of the rule is also translated into such formula.
3. consequently, by 1 and 2, any $F_{\leq}$-subtyping judgment is interpreted as a true formula of **SnS**, and, henceforth, $F_{\leq}$ is not essentially undecidable, i.e., possesses consistent decidable extensions; any **SnS**-translation of $F_{\leq}$ satisfying the above properties is such an extension.

It remains to show that the needed **SnS**-translations of $F_{\leq}$ with the above properties exist. We show it in the rest of this Section. The idea is quite simple: interpret type variables α, β, ... as corresponding **SnS**-atomic formulas $A(x)$, $B(x)$, ..., choosing a new predicate variable for each new type variable. Then knowing that $S(x)$ and $T(x)$ interpret σ and τ respectively, interpret:

- $\sigma \rightarrow \tau$ as $S(x) \supset T(x)$, or, more generally, as
$$S(x) \supset T(\mathbf{f}(x)),$$
- $\forall\alpha \leq \sigma \,.\, \tau$ as $\forall^2 A \left\{\forall^1 x[A(x) \supset S(x)] \supset T(x)\right\}$, or, more generally, as
$$\forall^2 A \left\{\forall^1 x[A(x) \supset S(x)] \supset T(\mathbf{g}(x))\right\},$$
where $\mathbf{f}$, $\mathbf{g}$ are arbitrary strings composed of **SnS**-successors.

Introduction of parameters $\mathbf{f}$ and $\mathbf{g}$ allows us to define *infinitely many different* interpretations of $F_{\leq}$ in **SnS**, see [18]. Surprising, but it works! We now proceed to formal definitions.

Definition 6.1 (SnS[$F_{\leq}$]($\mathbf{f}, \mathbf{g}$)-interpretations) *Let* $\mathbf{f}$ *and* $\mathbf{g}$ *be two arbitrary strings composed of successor function symbols of* **SnS**. *Both may be equal to the empty string* ε.

For an arbitrary type ρ *of* $F_{\leq}$, *the* Types-As-Propositions-Interpretation *of* ρ *in* **SnS** *with parameters* $\mathbf{f}$ *and* $\mathbf{g}$ *(the* **SnS**[$F_{\leq}$]($\mathbf{f}, \mathbf{g}$)*-interpretation for short) is defined as an* **SnS**-*formula* $[\![\, \rho \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x)$ *with unique distinguished free object variable* x *by induction on the structure of* ρ*:*

1. $[\![\, \alpha \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x) \quad \equiv_{df} \quad A(x)$ *(a new predicate variable A for each type variable α);*

2. $[\![\, \top \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x) \quad \equiv_{df} \quad x = x;$

3. $[\![\, \sigma \to \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x) \quad \equiv_{df} \quad [\![\, \sigma \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x) \supset [\![\, \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}}(\mathbf{f}(x));$

4. $[\![\, \forall\, \alpha \leq \sigma \,.\, \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x) \quad \equiv_{df}$

$$\forall^2 A \left\{ \forall^1 x \left(A(x) \supset [\![\, \sigma \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x) \right) \supset [\![\, \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}}(\mathbf{g}(x)) \right\}.$$

The $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$*-interpretation is extended to all subtyping judgments by:*

5. $[\![\, \sigma \leq \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}} \quad \equiv_{df} \quad \forall^1 x\, ([\![\, \sigma \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x) \supset [\![\, \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}}(x));$

6. $[\![\, \alpha_1 \leq \sigma_1 \ldots \alpha_n \leq \sigma_n \;\vdash\; \sigma \leq \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}} \quad \equiv_{df}$

$$[\![\, \alpha_1 \leq \sigma_1 \,]\!]^{\mathbf{f}}_{\mathbf{g}} \ldots [\![\, \alpha_n \leq \sigma_n \,]\!]^{\mathbf{f}}_{\mathbf{g}} \models_{\mathbf{SnS}} [\![\, \sigma \leq \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}}. \ \Box$$

Definition 6.2 (Theory) *Define the* $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$*-theory as:*

$$\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g}) \;\equiv_{df}\; \{\, \Gamma \vdash \sigma \leq \tau \mid [\![\, \Gamma \vdash \sigma \leq \tau \,]\!]^{\mathbf{f}}_{\mathbf{g}} \}$$

Further we will freely say that a typing judgment is true or valid in (or with respect to) a $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$*-interpretation iff it belongs to the set* $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$. □

Remarks. In $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$-interpretation we use just one-variable restricted fragment of **SnS**. If $\mathbf{f} = \mathbf{g} = \varepsilon$ then this fragment is also function-free (and can be seen as the *propositional second-order logic*). x is the only free object variable of any $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$-interpretation of any type. Subtyping judgments are interpreted as statements about **SnS**-semantical consequence relation $\models_{\mathbf{SnS}}$ containing no free object variables at all. Any $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$ is decidable. □

The **SnS**-interpretations enjoy the following important properties:

Lemma 6.3 (Embedding) *1) All axioms of* $F_{\leq}$ *are valid with respect to any* $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$.

2) All inference rules of $F_{\leq}$ *preserve validity with respect to any* $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$*, i.e., if both premises of a rule are valid in* $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$*, then so is the conclusion of the rule.* □

Proof . Straightforwardly rephrasing the proof from [18].

□

As a direct consequence we have, [18]:

Theorem 6.4 (On Decidable Extensions of $F_{\leq}$) *Any* $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$ *is a consistent decidable theory containing all* $F_{\leq}$*-derivable subtyping judgments. Henceforth,* $F_{\leq}$ *is not essentially undecidable possessing consistent decidable extensions.* □

Definition 6.5 ($F^{SnS}_{\leq}(\mathbf{f},\mathbf{g})$) *Define a system* $F^{SnS}_{\leq}(\mathbf{f},\mathbf{g})$ *as a combination of the inference rules from Figure 3 and a* $DECIDE$ *procedure for* $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$. □

Below, in Theorems 11.1 and 12.1 we show that all systems $F^{SnS}_{\leq}(\mathbf{f},\mathbf{g})$ also extend $F_{\leq}$ but are less coarse than **SnS**-interpretations, i.e.,

$$F_{\leq} \;\subset\; F^{SnS}_{\leq}(\mathbf{f},\mathbf{g}) \;\subset\; \mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g}) \qquad (4)$$

7 Consistency and Well-Structuredness of $F^{SnS}_{\leq}$

Proposition 7.1 *All systems* $F^{SnS}_{\leq}$ *are consistent: they do not prove, e.g.,* $\vdash \top \leq (\top \to \top)$*. Neither do they subtype any pair of differently structured types.* □

Proof . Immediate by definition. □

8 Inversion Principle

The main tool of the proofs of inclusions (4) (Theorems 11.1 and 12.1) and of the transitivity of $F^{SnS}_{\leq}(\mathbf{f},\mathbf{g})$ (Theorem 13.1) is the well-known inversion principle. The *rule invertibility* is the fundamental principle of the cut-free Gentzen-type derivation systems, see, e.g., [15].

The inversion principle is the key property needed to prove the minimal typing property for $F_{\leq}$. In fact, this is almost all what is needed to reconstruct $F_{\leq}$-inferences into normal forms, [8].

The inversion principle can be formulated as follows: *for an inference rule of a system* S

$$\frac{\Gamma \vdash \Phi \qquad\qquad \Gamma \vdash \Psi}{\Gamma \vdash \theta} \quad (Rule)$$

if a sequent $\Gamma \vdash \Theta$ *from the conclusion is derivable in* S *then the premises are also derivable in* S.

The inversion principle is important for goal-oriented proof-search procedures, which are guaranteed to be complete just stupidly applying inference rules bottom-up. Proofs in systems satisfying the inversion principle are direct, constructed from subproofs of subformulas of goal formulas, do not contain insights and roundabout ways.

The inversion principle is not evident, or even fails for systems with the CUT rule:

$$\frac{\Gamma \vdash A \supset C \qquad\qquad \Gamma \vdash C \supset B}{\Gamma \vdash A \supset B} \quad (Cut)$$

In the presence of (Cut), one cannot always be sure that a provable formula Θ of the form $A \supset B$ is obtained by some $(Rule)$ or by the (Cut). But applying (Cut) requires ingenuity to find intermediate formulas C, unattainable for mechanic theorem provers.

Note that the usual transitivity rule of $F_{\leq}$

$$\frac{\Gamma \vdash \tau_1 \leq \tau_2 \qquad \Gamma \vdash \tau_2 \leq \tau_3}{\Gamma \vdash \tau_1 \leq \tau_3} \quad (Trans)$$

has the definite (Cut) form.

Proposition 8.1 (Inversion for $F_{\leq}$, [8]) *In $F_{\leq}$ the rules $(Arrow)$ and (All) are invertible.* □

This may be seen as a good structural property.

9 Failure of the Inversion Principle for $\mathbf{SnS}[F_{\leq}](\mathbf{f}, \mathbf{g})$

The inversion principle **fails** for **SnS**-interpretations. In fact, we can have

$$[\![\Gamma \vdash (\sigma \to \tau) \leq (\sigma' \to \tau')]\!]^{\mathbf{f}}_{\mathbf{g}}$$

WITHOUT having

$$[\![\Gamma \vdash \sigma' \leq \sigma]\!]^{\mathbf{f}}_{\mathbf{g}} \text{ and } [\![\Gamma \vdash \tau \leq \tau']\!]^{\mathbf{f}}_{\mathbf{g}}$$

Take, for example, the judgment

$$\alpha \leq \top \vdash (\alpha \to \top) \leq (\top \to \top)$$

with the valid **SnS**-translation, but the **SnS**-translation of

$$\alpha \leq \top \vdash \top \leq \alpha$$

is false: $\forall^1 x(Ax \supset x = x) \not\models \forall^1 x(x = x \supset Ax)$.

10 Inversion Principle for $F^{SnS}_{\leq}$

Inversion principle trivially holds for $F^{SnS}_{\leq}$:

Lemma 10.1 (Inversion Principle) *In any $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$:*

- *if $\Gamma \vdash \sigma_1 \to \sigma_2 \leq \tau_1 \to \tau_2$ is provable, then $\Gamma \vdash \tau_1 \leq \sigma_1$ and $\Gamma \vdash \sigma_2 \leq \tau_2$ are also provable;*
- *if $\Gamma \vdash (\forall\alpha \leq \sigma_1 . \sigma_2) \leq (\forall\alpha \leq \tau_1 . \tau_2)$ is provable, then $\Gamma \vdash \tau_1 \leq \sigma_1$ and $\Gamma, \alpha \leq \tau_1 \vdash \sigma_2 \leq \tau_2$ are also provable.* □

Proof . Immediate by definition. In $F^{SnS}_{\leq}$ there are no other ways to subtype two $\to$- or $\forall$-types except applying $(Arrow)$ or (All) (or by the $(Refl)$, in which case the conclusion is straightforward). □

The proofs in $F^{SnS}_{\leq}$ are direct, one needs not subtype anything which do not belong to a goal subtyping judgment, proofs are conducted without roundabout ways and insights, completely deterministically.

11 $F^{SnS}_{\leq}$ is More Powerful than $F_{\leq}$

Now we prove two strict inclusions:

$$F_{\leq} \subset F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g}) \subset \mathbf{SnS}[F_{\leq}](\mathbf{f}, \mathbf{g})$$

So, the systems $F^{SnS}_{\leq}$ occupy an intermediate position between $F_{\leq}$ and **SnS**-interpretations: they are *more strong* than $F_{\leq}$ and *more subtle* than **SnS**-interpretations. Note that the decidable system $F^{\top}_{\leq}$ lies to the left of $F_{\leq}$ in the above diagram.

Remark. $F^{SnS}_{\leq}$ is an infinite family of systems. To decide normal forms each system uses a parametric $\mathbf{SnS}[F_{\leq}](\mathbf{f}, \mathbf{g})$-interpretation . For each $\mathbf{f}$ and $\mathbf{g}$ we have different parametric $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$. In fact, for the same $\mathbf{f}, \mathbf{g}$ we have the above inclusion $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g}) \subset \mathbf{SnS}[F_{\leq}](\mathbf{f}, \mathbf{g})$. In general, $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$ and $\mathbf{SnS}[F_{\leq}](\mathbf{f}', \mathbf{g}')$ are unrelated [18].

Theorem 11.1 ($F_{\leq} \subset F^{SnS}_{\leq}$) *Each system $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$ is strictly more powerful than $F_{\leq}$ is: if a subtyping judgment is provable in $F_{\leq}$ then it is also provable in $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$; the converse is not true in general.* □

Proof . See Appendix C.2. □

12 $F^{SnS}_{\leq}$ Are Less Coarse than SnS-Interpretations

We prove that $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$ subtypes strictly less types than the corresponding $\mathbf{SnS}[F_{\leq}](\mathbf{f}, \mathbf{g})$-interpretation:

Theorem 12.1 ($F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g}) \subset \mathbf{SnS}[F_{\leq}](\mathbf{f}, \mathbf{g})$.) *Each system $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$ is strictly less powerful than the corresponding interpretation $\mathbf{SnS}[F_{\leq}](\mathbf{f}, \mathbf{g})$: whatever is provable in $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$ is also true in $\mathbf{SnS}[F_{\leq}](\mathbf{f}, \mathbf{g})$; the converse in general does not hold. In particular, $F^{SnS}_{\leq}$ does not subtype differently structured types (e.g., a universally quantified and a functional type).* □

Proof . See Appendix C.3. □

13 Transitivity of $F^{SnS}_{\leq}$

Changing $F_{\leq}$ for $F^{SnS}_{\leq}$ we gain decidability and *do not lose* transitivity! Transitivity is an indispensable property needed for many purposes, in particular, for proof normalization, see [8, 21, 22].

Theorem 13.1 (Transitivity of $F^{SnS}_{\leq}$) *All systems $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$ are closed with respect to the transitivity rule $(Trans)$:*

whenever $\Gamma \vdash \sigma \leq \tau$ and $\Gamma \vdash \tau \leq \rho$ are provable in $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$, then $\Gamma \vdash \sigma \leq \rho$ is also provable in $F^{SnS}_{\leq}(\mathbf{f}, \mathbf{g})$.

Proof . See Appendix C.4. □

14 Improvements

The $F_{\leq}^{SnS}$-decision procedure may be obviously refined as follows: instead of pruning the $F_{\leq}^{Alg}$-proof tree[1] on the first application of (*Var-All-Decide*), one may fix $k \in \mathbb{N}$ and allow k applications of (*Var-All*) on each branch of a subtyping proof tree before applying (*Var-All-Decide*), which invokes the brute force **SnS**-decision procedure for normal forms. Denote the resulting system $F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})(k)$.

Consider a simple example. The non-modified procedure analyzing the normal form

$$\Gamma,\ \alpha \leq (\forall\beta(\top \to \top) \to \top)\ \vdash\ \alpha \leq (\forall\beta\,.\,\top \to \top)$$

returns $TRUE$. But if we allow just one application of (*Var-All*), we get $\Gamma ... \ \vdash\ (\top \to \top) \to \top\ \leq\ \top \to \top$, then $\Gamma\ \vdash\ \top\ \leq\ \top \to \top$, and, finally $FALSE$, which corresponds exactly to the $F_{\leq}$-proof.

With these modifications we still have for all $k \in \omega$

$$F_{\leq}\ \subset\ F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})(k)$$

It is not difficult to notice that

$$F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})(k+1)\ \subset\ F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})(k)$$

and $F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})(\infty)\ =\ F_{\leq}$.

The general theory of the converging sequences $\{\ F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})(k)\ \}_{k=0}^{\infty}$ is systematically developed in [21].

15 Conclusion

In this paper we concentrated exclusively on the the subtyping relations more powerful than in $F_{\leq}$. When combined with the usual $F_{\leq}$-term typing rules, our subtyping extensions produce systems, *which type strictly more terms* than $F_{\leq}$. Let $\Gamma\ \vdash\ \sigma \leq \tau$ be $F_{\leq}^{SnS}$-provable but $F_{\leq}$-unprovable. Then $\Gamma,\ x\ :\ \sigma,\ f\ :\ \tau \to \tau\ \ \vdash\ \ f\ x\ \ :\ \ \tau$ in $F_{\leq}^{SnS}$, but is untypable in $F_{\leq}$.

Therefore, the problems of subject reduction, strong normalization, and minimal typing are nontrivial for our extensions. If the general answers appear to be negative, it might be interesting to investigate restricted classes and/or to modify senses in which we understand the above properties. It would also be interesting to construct models of $F_{\leq}^{SnS}$. The work on these problems has been started [21, 22, 20].

As shows the example in Section 14, the systems $F_{\leq}^{SnS}$ (and hence **SnS**-interpretations) *do not separate* the sets of $F_{\leq}$-provable and $F_{\leq}$-finitely disprovable subtyping judgments. So, the problem is: whether these two sets are *recursively separable*. If yes, the separating cover of $F_{\leq}$ will be a better substitute for the $DECIDE$ component of the $F_{\leq}^{SnS}$-decision procedure.

In a particular case, when $\mathbf{f} = \mathbf{g} = \varepsilon$, our **SnS**-interpretations of $F_{\leq}$-subtyping are just interpretations into the *second-order propositional logic*. As it was established by Shamir [16], the class $PSPACE$ coincides with the class of languages recognizable by the so-called *interactive proof systems*. These systems are probabilistic algorithms exchanging messages in order to get convinced whether a given string belongs to a language with a given probability. It is challenging to introduce probabilistic algorithms in the domain of type systems.

[1] $F_{\leq}^{Alg}$ is the Curien-Ghelli algorithmic equivalent formulation of $F_{\leq}$, see [8] and Appendix B

Acknowledgments. I am greatly indebted to Luca Cardelli, Benjamin Pierce, Martín Abadi, Roberto Amadio, Philippe de Groote, Didier Galmiche, Jean-Luc Rémy, Hubert Comon, Michel Parigot for invaluable remarks, ideas, and discussions. I am grateful to my anonymous referee for his substantial help. To produce proof trees I used Paul Taylor's LaTeXmacro package. This work was done when I was at CRIN (Centre National de Recherche en Informatique de Nancy, France), which provided me the excellent research opportunities.

References

[1] Breazu-Tannen, V., Coquand, T., C., G., and Scedrov, A. Inheritance as implicit coercion. *Mathematical Structures in Computer Science 93* (1991), 172–221.

[2] Bruce, K. B., and Longo, G. A modest model of records, inheritance and bounded quantification. *Information and Computation 87* (1990), 196–240.

[3] Cardelli, L. A semantics of multiple inheritance. *Information and Computation 76* (1988), 138–164.

[4] Cardelli, L. Typeful programming. Research Report 45, Digital Equipment Corporation System Research Center, 1989.

[5] Cardelli, L., Martini, S., Mitchell, J., and Scedrov, A. An extension of system F with subtyping. *Information and Computation* (1994). To appear, preliminary version in LNCS'526, 1991, pp.550–570.

[6] Cardelli, L., and Wegner, P. On understanding types, data abstraction, and polymorphism. *Computing Surveys 17*, 4 (1985), 471–522.

[7] Castagna, G., and Pierce, B. C. Decidable bounded quantification. In *21st ACM Symp. on Principles of Programming Languages* (1994), pp. 151–162.

[8] Curien, P.-L., and Ghelli, G. Coherence of subsumption, minimum typing, and type checking in $F_{\leq}$. *Mathematical Structures in Computer Science 2* (1992), 55–91.

[9] Katiyar, D., and Sankar, S. Completely bounded quantification is decidable. In *ACM SIGPLAN Workshop on ML and its Applications* (1992).

[10] Leivant, D. Polymorphic type inference. In *10th ACM Symp. on Principles of Programming Languages* (1983), pp. 88–98.

[11] Mitchell, J. C. Type theory for programming languages. In *Handbook of Theoretical Computer Science* (1990), J. van Leeuwen, Ed., vol. B, Elsevier, pp. 365–458.

[12] Pierce, B. C. Bounded quantification is undecidable. In *19th ACM Symp. on Principles of Programming Languages* (1992), pp. 305–315.

[13] Rabin, M. Decidability of second order theories and automata on infinite trees. *Transactions of the American Mathematical Society 141* (1969), 1–35.

[14] Rabin, M. O. Decidable theories. In *Handbook of Mathematical Logic* (1977), J. Barwise, Ed., Studies in Logic and the Foundations of Mathematics, North Nolland, pp. 595–630.

[15] SCHWICHTENBERG, H. Proof theory: some applications of cut-elimination. In *Handbook of Mathematical Logic* (1977), J. Barwise, Ed., Studies in Logic and the Foundations of Mathematics, North-Holland Publishing Company, pp. 867–895.

[16] SHAMIR, A. PSPACE=IP. In *Proc. 31st IEEE FOCS* (1990), IEEE, pp. 11–15.

[17] TARSKI, A., MOSTOWSKI, A., AND ROBINSON, R. M. *Undecidable theories.* Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, 1953. Third printing, 1971.

[18] VOROBYOV, S. $F_{\leq}$: Bounded quantification is NOT essentially undecidable. Technical Report CRIN–94–R–018, Centre de Recherche en Informatique de Nancy, January 1994. FTP: FsubTAPI.[dvi,ps].Z .

[19] VOROBYOV, S. $F_{\leq}$ with recursive types: "Types-As-Propositions" Interpretations in M.Rabin's S2S. Technical Report CRIN–94–R–035, Centre de Recherche en Informatique de Nancy, February 1994. FTP: FsubREC.[dvi,ps].Z .

[20] VOROBYOV, S. β-η-$\top$-subject reduction and stucklessness for perfectly-structured second-order type systems with subtyping. In preparation, April –September 1994.

[21] VOROBYOV, S. Hierarchies of decidable extensions of bounded quantification. Research Report INRIA–RR–2354, Technical Report CRIN–94–R–120, Centre de Recherche en Informatique de Nancy, August –September 1994. FTP: INRIA–RR–2354.ps.Z, FsubHi.[dvi,ps].Z .

[22] VOROBYOV, S. Extensions of $F_{\leq}$ with Decidable Typing. Technical Report CRIN–94–R–127, Centre de Recherche en Informatique de Nancy, September 1994. FTP: FsubDecTyping.[dvi,ps].Z .

Note. Papers with mention `FTP: filename` are available by anonymous ftp **ftp.loria.fr** directory **pub/loria/prograis/vorobyov** or via URL **ftp://ftp.loria.fr/pub/loria/prograis/vorobyov** (mosaic).

A Monadic Second-Order Arithmetics

We briefly recall basic definitions and facts about *decidable* (weak) monadic second-order theories of one or several successors.

Fix arbitrary $n \in \omega \cup \{\omega\}$. The *alphabet* of n-successor monadic second-order arithmetic **SnS** consists of: 1) infinitely many object variables $x,\ y,\ z, \ldots$, 2) the equality predicate symbol $=$, 3) infinitely many unary (monadic) predicate variables $A,\ B,\ X,\ Y, \ldots$, 4) one, several, or countably many successor function symbols $\{succ_i\}_{i<n}$, 5) all usual boolean connectives, parentheses, 6) universal and existential first- and second-order quantifiers: $\forall^1$, $\exists^1$, $\forall^2$, $\exists^2$.

Terms are constructed as usual, starting from object variables by applying the successor function symbol(s).

Atomic formulas are either equalities of terms or expressions of the form $A(t)$, where A is a predicate variable and t is a term.

Formulas are constructed from atomic ones by the usual rules using boolean connectives, parentheses, first- and second-order quantifiers: $\forall^1 x\ \Phi$, $\exists^1 x\ \Phi$, $\forall^2 X\ \Phi$, $\exists^2 X\ \Phi$, (where x is an object and X is a predicate variable).

Interpretation. For an n-successor theory **SnS** consider the infinite n-ary tree T_n^∞. Interpret: 1) object variables as nodes of the tree, 2) $succ_i(t)$ as the i-th son of the node interpreting t, 3) equality, boolean connectives, and first-order quantifiers as usual, 4) predicate variables as arbitrary sets of nodes, 5) atomic formula $A(t)$ as the membership relation "the node t is in the set A"; 6) second-order quantifiers as quantifiers over sets of nodes.

Denote by $Th^2(\mathbf{SnS})$ or simply by **SnS** the set of all formulas valid under the above interpretation.

Replacing the interpretation 6) of the second-order quantifiers above by the following clause:

6′) second-order quantifiers are interpreted as quantifiers over *finite* sets of nodes,

we get the *weak monadic second order arithmetic of* n *successors*, denoted by **WSnS**.

All theories **WSnS** and **SnS** are *decidable.*

The most well known of all these are: Büchi's arithmetic **S1S**, Rabin's arithmetic **S2S**, and their weak counterparts **WS1S**, **WS2S**. The theory **S2S** is strictly more powerful than **WS2S**, **S1S**, and easily encodes all **SnS**. For details see [13, 14].

B $F_{\leq}^{Alg}$: Curien-Ghelli's Algorithmic Variant of $F_{\leq}$

Curien and Ghelli [8], Sect. 6.1, suggested $F_{\leq}^{Alg}$, an alternative equivalent formulation of $F_{\leq}$. We present it following [12]:

$$\Gamma \vdash \tau \leq \top \qquad (Top)$$

$$\Gamma \vdash \alpha \leq \alpha \ \ (\alpha\ is\ a\ variable) \qquad (Refl)$$

$$\frac{\Gamma \vdash \Gamma(\alpha) \leq \tau}{\Gamma \vdash \alpha \leq \tau} \qquad (AlgTrans)$$

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \quad \Gamma \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash \sigma_1 \to \sigma_2 \leq \tau_1 \to \tau_2} \qquad (Arrow)$$

$$\frac{\Gamma \vdash \tau_1 \leq \sigma_1 \quad \Gamma, \alpha \leq \tau_1 \vdash \sigma_2 \leq \tau_2}{\Gamma \vdash (\forall \alpha \leq \sigma_1 . \sigma_2) \leq (\forall \alpha \leq \tau_1 . \tau_2)} \qquad (All)$$

Three differences of $F_{\leq}^{Alg}$, as compared to $F_{\leq}$ are: 1) reflexivity $(Refl)$ is unlike $(Refl)$ of $F_{\leq}$ is restricted to variables, 2) transitivity $(Trans)$ is replaced by $(AlgTrans)$; 3) rules are applied in ordered manner (e.g., $(AlgTrans)$ never applies if $(Refl)$ is applicable).

Remark. Note that the inversion principle trivially holds for the $(Arrow)$ and (All) of $F_{\leq}^{Alg}$: a conclusion of each rule is provable *iff* so are the premises. Proofs in $F_{\leq}^{Alg}$ are direct, without roundabout ways.

Lemma B.1 ($F_{\leq}^{Alg} \equiv F_{\leq}$, [8]) *The systems $F_{\leq}$ and $F_{\leq}^{Alg}$ are equivalent: a subtyping judgment is derivable in $F_{\leq}$ iff it is derivable in $F_{\leq}^{Alg}$.* □

As an immediate consequence we have the following

Lemma B.2 (Inversion Principle for $F_{\leq}$) *In $F_{\leq}$:*

- *if* $\Gamma \vdash \sigma_1 \to \sigma_2 \leq \tau_1 \to \tau_2$ *is provable, then* $\Gamma \vdash \tau_1 \leq \sigma_1$ *and* $\Gamma \vdash \sigma_2 \leq \tau_2$ *are also provable;*
- *if* $\Gamma \vdash (\forall \alpha \leq \sigma_1 . \sigma_2) \leq (\forall \alpha \leq \tau_1 . \tau_2)$ *is provable, then* $\Gamma \vdash \tau_1 \leq \sigma_1$ *and* $\Gamma, \alpha \leq \tau_1 \vdash \sigma_2 \leq \tau_2$ *are also provable.*

Proof . Using equivalence of $F_{\leq}$ and $F_{\leq}^{Alg}$. Let $\Gamma \vdash \sigma_1 \to \sigma_2 \leq \tau_1 \to \tau_2$ be provable in $F_{\leq}$. Then it is provable in $F_{\leq}^{Alg}$. But the only way to prove it in $F_{\leq}^{Alg}$ consists in proving $\Gamma \vdash \tau_1 \leq \sigma_1$ and $\Gamma \vdash \sigma_2 \leq \tau_2$ in $F_{\leq}^{Alg}$ (since inversion principle holds for $F_{\leq}^{Alg}$). Henceforth, by equivalence, $\Gamma \vdash \tau_1 \leq \sigma_1$ and $\Gamma \vdash \sigma_2 \leq \tau_2$ are provable in $F_{\leq}$. The proof of the second claim is exactly the same. □

C Proofs

C.1 Proof of Proposition 4.2

Proof . (1). Is obvious. To prove (2) suppose, on the contrary, that a judgment of the form (2) is provable in $F_{\leq}$, i.e., there exists a proof, i.e, a sequence of judgments

$$J_0,\ J_1,\ \ldots,\ J_i,\ \ldots,\ J_n \equiv \Gamma \vdash \sigma \leq \alpha, \tag{5}$$

where each J_i is either an $F_{\leq}$-axiom, or is obtained from some J_k and J_l ($k < i$ and $l < i$) in the sequence by application of one of the $F_{\leq}$-rules: ($Arrow$), (All), or ($Trans$). Without loss of generality we can suppose that J_n is the first appearance of the judgment of the form (2) in the proof (5); otherwise, we can move left to select the first judgment of this form.

It remains to notice that J_n cannot be an axiom, since there are no $F_{\leq}$-axioms of the form (2). Next, neither ($Arrow$), nor (All) can produce a judgment of the form (2) (both produce types of the same structure). Therefore, (2) is obtained by ($Trans$). But to derive (2) by ($Trans$) one needs either $J_k \equiv \Gamma \vdash \sigma \leq \tau$ and $J_l \equiv \Gamma \vdash \tau \leq \alpha$ (τ non-variable type), or $J_k \equiv \Gamma \vdash \sigma \leq \beta$ and $J_l \equiv \Gamma \vdash \beta \leq \alpha$ (β is a type variable). Therefore, J_k has the form (2) and appears in (5) before J_n. But this contradicts to the choice of J_n. □

C.2 Proof of Theorem 11.1

Let a subtyping judgment $J \equiv \Gamma \vdash \sigma \leq \tau$ be provable in $F_{\leq}$. Then, by equivalence of $F_{\leq}$ and $F_{\leq}^{Alg}$ (Lemma B.1), it is provable in $F_{\leq}^{Alg}$. Consider the $F_{\leq}^{Alg}$-inference tree of J. If this tree does not contain applications of the rule ($AlgTrans$) corresponding to the (Var-All-$Decide$) rule, then this tree is also the $F_{\leq}^{SnS}$-inference tree of J and we are done.

Suppose now that the $F_{\leq}^{Alg}$-inference tree $\mathcal{T}$ of J *does contain* applications of ($AlgTrans$) corresponding to the (Var-All-$Decide$) rule. Transform this tree $\mathcal{T}$ as follows. Starting from the root J follow each branch till the first application of ($AlgTrans$) (if any), and cut it on this application so as the conclusion of ($AlgTrans$) remains in the tree. Denote by $\mathcal{T}'(J_1,\ldots,J_n)$ the resulting tree, where $J_1,\ldots,J_n$ are all leaves-conclusions of ($AlgTrans$) remaining after the above pruning. Note that $\mathcal{T}'(J_1,\ldots,J_n)$ is exactly the $F_{\leq}^{SnS}$-inference tree, and $J_1,\ldots,J_n$ are precisely $F_{\leq}^{SnS}$-normal forms. Instead of applying ($AlgTrans$), the $F_{\leq}^{SnS}$-decision procedure transforms $J_1,\ldots,J_n$ into **SnS**-formulas and decides them. So, to finish our proof we have to prove that $J_1,\ldots,J_n$ are interpreted as true **SnS**-formulas.

To do this, notice, that by equivalence of $F_{\leq}$ and $F_{\leq}^{Alg}$, all the judgments $J_1,\ldots,J_n$ are provable in $F_{\leq}$. But by Theorem 6.4 above everything provable in $F_{\leq}$ is true with respect to any **SnS**-interpretation.

The strictness of inclusion is simple: since $F_{\leq}^{SnS}$ is decidable and $F_{\leq}$ is not, there should certainly exist $F_{\leq}^{SnS}$-provable and not $F_{\leq}$-provable subtyping judgments. □

C.3 Proof of Theorem 12.1

Again applying Theorem 6.4 above, all $F_{\leq}^{SnS}$-inference rules preserve validity with respect to any **SnS**-interpretation. As normal forms of $F_{\leq}^{SnS}$ are decided by the same **SnS**-decision procedure, they are simultaneously true with respect to an **SnS**-interpretation $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$ and $F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})$. By definition, $F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})$ does not subtype differently structured types, whereas **SnS**-interpretations do, e.g., $\vdash \top \to \top \leq \forall\alpha.\top$ is true in any **SnS**-interpretation. □

C.4 Proof of Theorem 13.1

By induction on complexity of subtyping inference.

Suppose the premises of the theorem hold, i.e.,
$\Gamma \vdash \sigma \leq \tau$ and $\Gamma \vdash \tau \leq \rho$ are $F_{\leq}^{SnS}$-provable.

We must show that so is $\Gamma \vdash \sigma \leq \rho$.

We have to consider several cases:

1. ρ is $\top$;
2. ρ is a type variable;
3. ρ is an arrow or a universal type, both σ and τ are type variables;
4. τ and ρ are both arrow types and σ is a type variable;
5. τ and ρ are both universal types and σ is a type variable;
6. σ, τ, and ρ are all arrow types;
7. σ, τ, and ρ are all universal types.

Case 1. Vacuous: $\Gamma \vdash \sigma \leq \top$, always.

Case 2. If ρ is a type variable then σ and τ should also be type variables; otherwise the rule ($TVar$-R-2) would disprove one of the premises of the theorem.

So we should demonstrate that $F_{\leq}^{SnS}$-provability of:

$$\Gamma \vdash \alpha \leq \beta, \tag{6}$$
$$\Gamma \vdash \beta \leq \gamma \tag{7}$$

imply the $F_{\leq}^{SnS}$-provability of

$$\Gamma \vdash \alpha \leq \gamma \tag{8}$$

for type variables α, β, γ.

Note that the $F_{\leq}^{SnS}$-proofs of (6) and (7) are just finite sequences of ($TVar$-R-1)-applications finishing by an application of ($Refl$). These two sequences could be easily merged into just one such sequence proving (8). Indeed, starting from the judgment (8) by backward applications of ($TVar$-R-1) we are guaranteed (by provability of (6)) to reach β on the left of $\leq$, i.e., we reach (7), which is provable by hypothesis.

Case 3. Suppose that ρ is either an $\to$- or a $\forall$-type, σ and τ are type variables α and β respectively.

We transform the proofs of

$$\Gamma \vdash \alpha \leq \beta, \tag{9}$$
$$\Gamma \vdash \beta \leq \rho \tag{10}$$

into the proof of

$$\Gamma \vdash \alpha \leq \rho \tag{11}$$

as follows. Starting from the judgment (11) we first repeat (backwards) exactly the same sequence of steps as in the proof of (9), which leads to $\Gamma \vdash \beta \leq \beta$ (but applying (*Var-Arrow*) or (*Var-All*-1) instead of (*TVar-R*-1)). This gives the inference of (11) from (10) used as axiom. We then repeat the proof of the latter judgment, which exists by assumption. The result is the desired proof.

Case 4. Suppose

$$\Gamma \vdash \alpha \quad \leq \quad \tau_1 \to \tau_2, \tag{12}$$

$$\Gamma \vdash \tau_1 \to \tau_2 \quad \leq \quad \rho_1 \to \rho_2 \tag{13}$$

are $F_{\leq}^{SnS}$-provable. We must prove that so is

$$\Gamma \vdash \alpha \leq \rho_1 \to \rho_2 \tag{14}$$

The proof of (12) is a finite sequence of (*Var-Arrow*) followed either a) by (*Refl*) or b) by (*Arrow*).

In the Case 4.a we construct the proof of (14) (in a backward manner) first applying to (14) exactly the same sequence of (*Var-Arrow*) applications until (*Refl*), as in the proof of (12). This gives a subinference of (14) from (13) used as an axiom. We then complete the latter subinference by including the proof of (13) (which is $F_{\leq}^{SnS}$-provable by assumption).

In the Case 4.b we construct the proof of (14) as follows. Considering the final part of the inference of (12) till the first application of (*Arrow*):

$$\cfrac{\Gamma \vdash \sigma_1 \to \sigma_2 \leq \tau_1 \to \tau_2 \quad (\surd)}{\cfrac{\Gamma \vdash \alpha' \leq \tau_1 \to \tau_2}{\cfrac{\vdots}{\alpha \leq \tau_1 \to \tau_2}}} \tag{15}$$

we see that (12) is provable iff $(\surd)$ is provable. By the inversion property for $F_{\leq}^{SnS}$ (Theorem 10.1) this implies provability of

$$\Gamma \vdash \tau_1 \quad \leq \quad \sigma_1, \tag{16}$$

$$\Gamma \vdash \sigma_2 \quad \leq \quad \tau_2 \tag{17}$$

Similarly, provability of (13) implies provability of

$$\Gamma \vdash \rho_1 \quad \leq \quad \tau_1, \tag{18}$$

$$\Gamma \vdash \tau_2 \quad \leq \quad \rho_2 \tag{19}$$

Applying the inductive hypothesis to (18) and (16), then to (17) and (19) we get the $F_{\leq}^{SnS}$-provability of $\Gamma \vdash \rho_1 \leq \sigma_1$ and $\Gamma \vdash \sigma_2 \leq \rho_2$.

But this means that $\sigma_1 \to \sigma_2 \leq \rho_1 \to \rho_2$ is also $F_{\leq}^{SnS}$-provable. This allows us to transform the proof (15) into the proof of (14) by simple replacement of $\tau_1 \to \tau_2$ by $\rho_1 \to \rho_2$.

Case 5. Suppose

$$\Gamma \vdash \alpha \quad \leq \quad (\forall\beta \leq \tau_1 \,.\, \tau_2), \tag{20}$$

$$\Gamma \vdash (\forall\beta \leq \tau_1 \,.\, \tau_2) \quad \leq \quad (\forall\beta \leq \rho_1 \,.\, \rho_2) \tag{21}$$

are $F_{\leq}^{SnS}$-provable. We have to prove that

$$\Gamma \vdash \alpha \leq (\forall\beta \leq \rho_1 \,.\, \rho_2) \tag{22}$$

The proof of (20) is a finite (possibly empty) sequence of (*Var-All*-1) followed either a) by (*TVar*) or b) by (*Var-All-Decide*).

In the Case 5.a we construct the proof of (22) first applying to it the same sequence of (*Var-All*-1) as in the proof of (20), until (*TVar*). This gives a subinference of (22) from (21) used as axiom. We then complete the latter subinference by including the proof of (21) (which is $F_{\leq}^{SnS}$-provable by assumption).

In the Case 5.b we construct the proof of (22) as follows. Consider the final part of the inference of (20) till the application of (*Var-All-Decide*):

$$\cfrac{\Gamma \vdash DECIDE(\Gamma \vdash \alpha' \leq \forall\beta \leq \tau_1 \,.\, \tau_2)(\surd)}{\cfrac{\Gamma \vdash \alpha' \leq (\forall\beta \leq \tau_1 \,.\, \tau_2)}{\cfrac{\vdots}{\alpha \leq (\forall\beta \leq \tau_1 \,.\, \tau_2)}}} \tag{23}$$

We see that (20) is provable iff the $F_{\leq}^{SnS}$-normal form in $(\surd)$ is valid in a chosen theory $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$. As each $\mathbf{SnS}[F_{\leq}](\mathbf{f},\mathbf{g})$ is more powerful than the corresponding $F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})$ (Theorem 12.1), the $F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})$-provability of (20) implies that:

$$[\![\, \Gamma \,]\!]_{\mathbf{g}}^{\mathbf{f}} \models_{\mathbf{SnS}} \forall^1 x[A'(x) \supset [\![\, \forall\beta \leq \tau_1 \,.\, \tau_2 \,]\!]_{\mathbf{g}}^{\mathbf{f}}(x)] \tag{24}$$

Similarly, the $F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})$-provability of (21) implies

$$[\![\, \Gamma \,]\!]_{\mathbf{g}}^{\mathbf{f}} \models_{\mathbf{SnS}} \forall^1 x[[\![\, \forall\beta \leq \tau_1 \,.\, \tau_2 \,]\!]_{\mathbf{g}}^{\mathbf{f}}(x) \supset [\![\, \forall\beta \leq \rho_1 \,.\, \rho_2 \,]\!]_{\mathbf{g}}^{\mathbf{f}}(x)] \tag{25}$$

Henceforth, by syllogistics, (24) and (25) imply

$$[\![\, \Gamma \,]\!]_{\mathbf{g}}^{\mathbf{f}} \models_{\mathbf{SnS}} \forall^1 x[A'(x) \supset [\![\, \forall\beta \leq \rho_1 \,.\, \rho_2 \,]\!]_{\mathbf{g}}^{\mathbf{f}}(x)] \tag{26}$$

Now, to construct the inference of (22) we start by the sequence of the same (*Var-All*) applications as in (23) till $\Gamma \vdash \alpha' \leq (\forall\beta \leq \rho_1 \,.\, \rho_2)$. After that we should apply either the rule (*TVar*) (in this case we are done), or the rule (*Var-All-Decide*) getting $DECIDE(\Gamma \vdash \alpha' \leq (\forall\beta \leq \rho_1 \,.\, \rho_2))$. But in the latter case $DECIDE$ should necessarily return the result $TRUE$ (by (26)), and the desired $F_{\leq}^{SnS}(\mathbf{f},\mathbf{g})$-proof is completed.

Case 7. Let

$$\Gamma \vdash (\forall\beta \leq \sigma_1 \,.\, \sigma_2) \quad \leq \quad (\forall\beta \leq \tau_1 \,.\, \tau_2), \tag{27}$$

$$\Gamma \vdash (\forall\beta \leq \tau_1 \,.\, \tau_2) \quad \leq \quad (\forall\beta \leq \rho_1 \,.\, \rho_2) \tag{28}$$

We have to show

$$\Gamma \vdash (\forall\beta \leq \sigma_1 \,.\, \sigma_2) \leq (\forall\beta \leq \rho_1 \,.\, \rho_2) \tag{29}$$

By Inversion principle (Lemma 10.1) from (27) and (28) we get:

$$\Gamma \vdash \tau_1 \quad \leq \quad \sigma_1 \tag{30}$$

$$\Gamma, \beta \leq \tau_1 \vdash \sigma_2 \quad \leq \quad \tau_2 \tag{31}$$

$$\Gamma \vdash \rho_1 \quad \leq \quad \tau_1 \tag{32}$$

$$\Gamma, \beta \leq \rho_1 \vdash \tau_2 \quad \leq \quad \rho_2 \tag{33}$$

From (32) and (30) by induction hypothesis we get

$$\Gamma \vdash \rho_1 \leq \sigma_1 \tag{34}$$

From (31), (32) and (33) by induction hypothesis we get

$$\Gamma,\ \alpha \leq \rho_1 \vdash \sigma_2 \leq \rho_2 \tag{35}$$

(each time instead of using the hypothesis $\beta \leq \tau_1$ we use the hypothesis $\beta \leq \rho_1$ and (32)). But (34) and (35) imply (29).

Case 6 is completely analogous to the preceding one. □

INFORMATIK

Below you find a list of the most recent technical reports of the research group *Logic of Programming* at the Max-Planck-Institut für Informatik. They are available by anonymous ftp from our ftp server `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Regina Kraemer
Im Stadtwald
D-66123 Saarbrücken
GERMANY
e-mail: `kraemer@mpi-sb.mpg.de`

MPI-I-94-246	M. Hanus	On Extra Variables in (Equational) Logic Programming
MPI-I-94-241	J. Hopf	Genetic Algorithms within the Framework of Evolutionary Computation: Proceedings of the KI-94 Workshop
MPI-I-94-240	P. Madden	Recursive Program Optimization Through Inductive Synthesis Proof Transformation
MPI-I-94-239	P. Madden, I. Green	A General Technique for Automatically Optimizing Programs Through the Use of Proof Plans
MPI-I-94-238	P. Madden	Formal Methods for Automated Program Improvement
MPI-I-94-235	D. A. Plaisted	Ordered Semantic Hyper-Linking
MPI-I-94-234	S. Matthews, A. K. Simpson	Reflection using the derivability conditions
MPI-I-94-233	D. A. Plaisted	The Search Efficiency of Theorem Proving Strategies: An Analytical Comparison
MPI-I-94-232	D. A. Plaisted	An Abstract Program Generation Logic
MPI-I-94-230	H. J. Ohlbach	Temporal Logic: Proceedings of the ICTL Workshop
MPI-I-94-229	Y. Dimopoulos	Classical Methods in Nonmonotonic Reasoning
MPI-I-94-228	H. J. Ohlbach	Computer Support for the Development and Investigation of Logics
MPI-I-94-226	H. J. Ohlbach, D. Gabbay, D. Plaisted	Killer Transformations
MPI-I-94-225	H. J. Ohlbach	Synthesizing Semantics for Extensions of Propositional Logic
MPI-I-94-224	H. Aït-Kaci, M. Hanus, J. J. M. Navarro	Integration of Declarative Paradigms: Proceedings of the ICLP'94 Post-Conference Workshop Santa Margherita Ligure, Italy
MPI-I-94-223	D. M. Gabbay	LDS – Labelled Deductive Systems: Volume 1 — Foundations
MPI-I-94-218	D. A. Basin	Logic Frameworks for Logic Programs
MPI-I-94-216	P. Barth	Linear 0-1 Inequalities and Extended Clauses
MPI-I-94-209	D. A. Basin, T. Walsh	Termination Orderings for Rippling