Polymorphic Subtyping Without Distributivity

Jacek Chrząszcz

10th March 1998

Abstract

The subtyping relation in the polymorphic second-order λ-calculus was introduced by John C. Mitchell in 1988. It is known that this relation is undecidable, but all known proofs of this fact strongly depend on the distributivity axiom. Nevertheless it has been conjectured that this axiom does not influence the undecidability.

The present paper shows undecidability of subtyping when we remove distributivity from its definition. Furthermore, the full equational axiomatisation of the corresponding equivalence relation is given. Both results follow from an analysis of rewriting-style subtyping derivations.

1 Introduction

Polymorphism is one of the most important issues in modern programming. When we go beyond the limits of shallow polymorphism imposed by common functional languages like ML or Haskell, we encounter (among others) the following question: when can values of a given type σ replace values of type τ without a type clash, or, differently speaking, is σ a *subtype* of τ?

In the higher-order polymorphic functional language, system **F**, the subtyping relation is characterized by:

$$\forall\alpha_1 \ldots \alpha_n\, \sigma \;\sqsubseteq_{\mathbf{F}}\; \forall\beta_1 \ldots \beta_m\, \sigma\{\alpha_1 \mapsto \rho_1, \ldots, \alpha_n \mapsto \rho_n\} \tag{1}$$

where $\beta_1, \ldots, \beta_m$ are not free in $\forall\alpha_1 \ldots \alpha_n\, \sigma$, and $\rho_1, \ldots, \rho_n$ are arbitrary types. But this relation is not strong enough. One can see for example that when $\sigma \sqsubseteq_{\mathbf{F}} \sigma'$ then terms of type $\tau \to \sigma$ can very well replace those of type $\tau \to \sigma'$. Therefore John C. Mitchell in [Mit88] extended the notion of subtyping to a containment relation $\sqsubseteq_{\mathrm{M}}$. It is defined by four axioms and three rules. The axioms are: reflexivity, quantifier instantiation, generalization (adding an empty quantifier), and quantifier distributivity over an arrow. The rules ensure transitivity and a particular closure by context. The particularity lies within the rule concerning the arrow symbol:

$$\sigma_1 \to \tau_1 \sqsubseteq_{\mathrm{M}} \sigma_2 \to \tau_2 \qquad \text{if} \qquad \sigma_2 \sqsubseteq_{\mathrm{M}} \sigma_1 \quad \text{and} \quad \tau_1 \sqsubseteq_{\mathrm{M}} \tau_2$$

Adding a subsumption rule for $\sqsubseteq_{\mathrm{M}}$

$$\frac{E \vdash M : \sigma \quad \sigma \sqsubseteq_{\mathrm{M}} \tau}{E \vdash M : \tau}$$

properly extends System $\mathbf{F}$. Unfortunately the resulting system $\mathbf{F}_\eta$ has all the bad undecidability properties of system $\mathbf{F}$. Typability and type-checking are undecidable in both systems (see [Wel94] for $\mathbf{F}$ and [Wel96] for typability and [TU95] for type-checking in $\mathbf{F}_\eta$) and so is Mitchell subtyping relation [TU95] (see also [Wel95] for alternative proof).

The solution for designers of functional languages with full polymorphism is to seek for non-trivial decidable subrelations of $\sqsubseteq_{\mathrm{M}}$. And to do this it is essential to understand what exactly makes this relation undecidable.

The present paper analyses the subtyping relation when we remove distributivity from its definition. This axiom seems to be the least intuitive while the subtyping relation without it remains quite powerful and useful.

Both known undecidability proofs strongly depend on the distributivity axiom. Tiuryn and Urzyczyn [TU95] encode computations of a certain device called *stack register machine* with undecidable halting problem. The encoding is based on subtyping derivations in Longo-Milsted-Soloviev system [LMS95], which deeply incorporates the distributivity axiom.

Another approach is taken by Wells [Wel95], who reduces the well known semi-unification problem to subtyping. Having two pairs of types σ_1, τ_1 and σ_2, τ_2, the author constructs a universal type $\Pi(\delta_1, \zeta_1, \delta_2, \zeta_2)$ such that if $\vec{\gamma}$ are all variables in $\sigma_1, \tau_1, \sigma_2, \tau_2$ then

$$\forall\beta\,(\Pi(\beta,\beta,\beta,\beta) \to \bot) \;\sqsubseteq_{\mathrm{M}}\; (\forall\vec{\gamma}\,\Pi(\sigma_1,\tau_1,\sigma_2,\tau_2)) \to \bot$$

if and only if σ_1, τ_1 and σ_2, τ_2 are semi-unifiable. The distributivity axiom plays a crucial role in this proof. Intuitively after performing common substitution, it is the only way to separate quantifiers and therefore apply two separate substitutions yielding $\sigma_1\theta\theta_1 = \tau_1\theta$ and $\sigma_2\theta\theta_2 = \tau_2\theta$.

Our work proves that removing distributivity does not suffice to obtain a decidable subtyping relation.

The undecidablility result is based on a reduction similar to that in the work of Tiuryn and Urzyczyn. Instead of using Longo-Milsted-Soloviev system we define a type rewriting system (like in [Tiu95]) which is a straightforward translation of original Mitchell axioms. Each rewriting step is an application of instantiation or generalisation axioms, some of them being applications backwards because of the symbol $\to$ which is anti-monotonic in the first argument.

The rewriting and its correspondence with the subtyping axiomatisation is given in Section 3. Then it is shown that rewriting derivations can be normalized, i.e. the steps can be rearranged so that all positive steps precede the negative ones. Using normal derivations we characterize in Section 6 the equivalence relation induced by the new subtyping preorder. Its equational axiomatization is given, similar to the one given by Tiuryn [Tiu95] for the full Mitchell subtyping relation.

Section 7 introduces yet another class of derivations, ordered ones, which are normal derivations where rewriting takes place from the root towards the leaves in the positive phase, and the other way in the negative ones. It is somewhat similar to Wells's syntax-driven system [Wel95], but using rewriting helps keeping all proof information in one piece.

Ordered derivations constitute a very useful tool to prove subtyping properties necessary to encode computations of a stack register machine (Section 9). Finally, in Section 10 we conclude the paper with some sugestions on how else a decidable subtyping relation could be found.

2 Notation

We work with polymorphic types (like in system **F**), defined over the denumerable set of variables V by the following grammar:

$$\mathcal{T} ::= V \mid \forall V\, \mathcal{T} \mid \mathcal{T} \to \mathcal{T}$$

Variables appearing in types are denoted by the initial Greek letters α, β, γ, sequences of quantified variables by $\vec{\delta}$, $\vec{\zeta}$, $\vec{\vartheta}$, and types by other Greek letters like σ, τ, ρ... Often we abbreviate $\forall\alpha\forall\beta\,\tau$ as $\forall\alpha\beta\,\tau$. We admit that quantifiers bind stronger than the operator $\to$, and we use the symbol $\bot$ to denote the type $\forall\alpha\,\alpha$.

The number of arrows appearing in a given type is called its *weigth*. The inductive definition follows:

$$\begin{aligned} \mathit{weight}(\alpha) &= 0 \\ \mathit{weight}(\forall\alpha\,\sigma) &= \mathit{weight}(\sigma) \\ \mathit{weight}(\sigma_1 \to \sigma_2) &= \mathit{weight}(\sigma_1) + \mathit{weight}(\sigma_2) + 1 \end{aligned}$$

Because of quantifiers, variables occuring in a given type may be either bound or free, that is not bound by any quantifier. For a given type σ, the set of all free variables occuring in it is denoted by $FV(\sigma)$. Formally we define:

$$\begin{aligned} FV(\alpha) &= \{\alpha\} \\ FV(\sigma_1 \to \sigma_2) &= FV(\sigma_1) \cup FV(\sigma_2) \\ FV(\forall\beta\,\sigma) &= FV(\sigma) \setminus \{\beta\} \end{aligned}$$

Types which differ only in the names of bound variables are considered to be equal. Furthermore we assume that in each type the names of bound variables are distinct and different from the names of free ones.

A partial function of a finite domain from the set of variables V to the set of types $\mathcal{T}$ is called a *substitution*. It can be extended in a natural way to a function from $\mathcal{T}$ to $\mathcal{T}$. Application of a substitution θ to a type σ is written in postfix notation as $\sigma\theta$. If θ assigns a type τ_i to variables α_i for all i and is undefined for other variables, we write $\theta = \{\alpha_1 \mapsto \tau_1, \ldots, \alpha_n \mapsto \tau_n\}$. If necessary we rename bound variables in σ to ensure that free variables in the inserted types $\tau_1, \ldots, \tau_n$ remain free in $\sigma\theta$. In our notation substitutions bind stronger than quantifiers, so in the type $\forall\beta\,\sigma\{\alpha \mapsto \xi\}$ the free occurences of β in the type ξ will be bound by the quantifier $\forall\beta$.

Polimorphic types may also be regarded as labelled trees. Leaves are labelled by variables, internal nodes by $\to$ symbols and each node is labelled by a finite sequence of quantifiers. The set of positions in such a tree can be identified

with a finite set of finite chains over the alphabet $\{0, 1\}$. We define the sign of a given position p to be positive if p contains even number of zeros, and negative otherwise. The empty chain is denoted by ϵ, and given two chains p, r their concatenation is writen as pr.

Positions can be compared by two different partial orders. The prefix order is denoted $\leq$, and the notation $p \not\gtrless r$ means that p and r are not comparable. The other order is the lexicographic order $\preceq$ induced by $0 \prec 1$.

If σ is a type and p a position in the tree of σ, then we denote by $\sigma|_p$ a subtree at the position p. If σ and ϱ are types and p is a position in the tree of σ then $\sigma[\varrho]_p$ stands for σ with the subtree at position p replaced by ϱ. Unlike a substitution, this operation may make some free variables in ϱ bound by quantifiers in σ.

3 Subtyping relation

The relation of subtyping $\sqsubseteq$ is defined by 3 axioms and 3 rules:

$$\begin{array}{rl} \textbf{(refl)} & \sigma \sqsubseteq \sigma \\ \textbf{(inst)} & \forall\alpha\,\sigma \sqsubseteq \sigma\{\alpha \mapsto \rho\} \\ \textbf{(quant)} & \sigma \sqsubseteq \forall\alpha\,\sigma \quad \text{if } \alpha \notin FV(\sigma) \end{array}$$

Rules:

$$\begin{array}{rl} \textbf{(}\rightarrow\textbf{-context)} & \dfrac{\sigma_2 \sqsubseteq \sigma_1 \quad \tau_1 \sqsubseteq \tau_2}{\sigma_1 \rightarrow \tau_1 \sqsubseteq \sigma_2 \rightarrow \tau_2} \\ \textbf{(}\forall\textbf{-context)} & \dfrac{\sigma \sqsubseteq \tau}{\forall\alpha\,\sigma \sqsubseteq \forall\alpha\,\tau} \\ \textbf{(trans)} & \dfrac{\sigma \sqsubseteq \rho \quad \rho \sqsubseteq \tau}{\sigma \sqsubseteq \tau} \end{array}$$

Our definiton differs from the original one introduced by Mitchell in [Mit88] in the lack of the distributivity axiom:

$$\textbf{(distr)} \qquad \forall\alpha(\sigma \rightarrow \tau) \sqsubseteq_{\mathrm{M}} \sigma \rightarrow \forall\alpha\,\tau \quad \text{if } \alpha \notin FV(\sigma)$$

4 Type rewriting

For technical reasons instead of the derivations in the above system we use type rewriting according to the following rules:

$$\begin{array}{llll} C^+[\forall\alpha\,\sigma]_p & \overset{p}{\Longrightarrow}{}^+_i & C^+[\sigma\{\alpha \mapsto \rho\}]_p & \\ C^+[\sigma]_p & \overset{p}{\Longrightarrow}{}^+_q & C^+[\forall\alpha\,\sigma]_p & \text{if } \alpha \notin FV(\sigma) \end{array} \qquad \begin{array}{llll} C^-[\sigma\{\alpha \mapsto \rho\}]_r & \overset{r}{\Longrightarrow}{}^-_i & C^-[\forall\alpha\,\sigma]_r & \\ C^-[\forall\alpha\,\sigma]_r & \overset{r}{\Longrightarrow}{}^-_q & C^-[\sigma]_r & \text{if } \alpha \notin FV(\sigma) \end{array}$$

where $C^+[\,]_p$ and $C^-[\,]_r$ denote contexts with a hole at a positive position p and negative position r respectively. When we do not want to precise what step

is considered, we omit the appropriate decoration and write for example $\Longrightarrow_i$ or $\Longrightarrow^+$. Any finite number of steps may be denoted by a double arrow $\Longrightarrow\!\!\!\!\Rightarrow$. Given a derivation step $\sigma \to \tau$, the absolute value of the difference of weights of σ and τ is called the *weight* of this step. Intuitively, it is the number of arrows introduced (or removed) by this step.

The meaning of the introduced rewriting is expressed by the following lemma:

Lemma 4.1 *For any types σ and τ we have:*

$$\sigma \sqsubseteq \tau \quad \textit{if and only if} \quad \sigma \Longrightarrow\!\!\!\!\Rightarrow \tau$$

Proof. Both implications can easily be proved by a routine induction. ■

5 Normal derivations

The goal of the present section is to show that every subtyping inequality can be proved by a *normal* derivation, that is a sequence of positive steps (which increase the weight of a type) followed by a sequence of negative steps.

Definition 5.1 `(normal derivation)`
A derivation $\sigma \Longrightarrow\!\!\!\!\Rightarrow \tau$ is called normal, *if it is of the form $\sigma \Longrightarrow\!\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\!\Rightarrow^- \tau$ for certain type ρ.*

The *weight* of a normal derivation is the sum of weights of steps in it. It is easy to see that it is equal $2 \times \mathit{weight}(\rho) - \mathit{weight}(\sigma) - \mathit{weight}(\tau)$.

Before we show how to normalise a given derivation, we define the notion of safe steps.

Definition 5.2 `(safe step)`
A rewriting step $\sigma \Longrightarrow \tau$ is safe, *if one of the following conditions is satisfied:*

1. *it is a $\Longrightarrow_q$ step;*
2. *it is a $\Longrightarrow_i$ step, in which the inserted type is $\bot$;*
3. *it is a $\overset{p}{\Longrightarrow}_i$ step, in which the inserted type is a variable β bound at a position $p' \leq p$, of the same sign as the sign of p.*

The second condition above covers particularly the instatiation of empty quantifiers, i.e. the steps $C^+[\forall\alpha\,\sigma] \Longrightarrow_i^+ C^+[\sigma]$ and respectively $C^-[\sigma] \Longrightarrow_i^- C^-[\forall\alpha\,\sigma]$ for $\alpha \notin FV(\sigma)$.

The last condition of the safe step definition expresses the following situations:

$$\begin{aligned} C^+[\forall\beta\; C^+[\forall\alpha\,\sigma]] \quad &\Longrightarrow_i^+ \quad C^+[\forall\beta\; C^+[\sigma\{\alpha \mapsto \beta\}]] \\ C^-[\forall\beta\; C^+[\sigma\{\alpha \mapsto \beta\}]] \quad &\Longrightarrow_i^- \quad C^-[\forall\beta\; C^+[\forall\alpha\,\sigma]] \end{aligned}$$

All safe steps have the weight 0. In addition, a safe step can be inserted in the derivation without changing its weight. The formal statement of this property,

together with an example of a step of weight 0 which is not safe will be given by the end of the section.

The following lemma shows how to permute negative and positive steps,[1] which is necessary to normalise a derivation.

Lemma 5.3 (permuting negative and positive steps)
Let σ and τ be types such that $\sigma \Longrightarrow^- \rho \Longrightarrow^+ \tau$. There exists a type ρ' such that $\sigma \Longrightarrow^+ \rho' \Longrightarrow^- \tau$. In addition if the step $\sigma \Longrightarrow^- \rho$ is safe then after the permutation the step $\rho' \Longrightarrow^- \tau$ will remain safe and similarily if $\rho \Longrightarrow^+ \tau$ is a safe step then so will be $\sigma \Longrightarrow^+ \rho'$.

Proof. Let $\sigma \overset{p}{\Longrightarrow}^- \rho \overset{r}{\Longrightarrow}^+$. Depending on p and r we have three possibilities to consider:

- $p \not\gtrless r$ — this case is easy;
- $p < r$, $r = ps$. If both steps are q-steps then it is obvious. If we have one q-step and one i-step, then it does not pose any problems either. The type inserted by the i-step is unchanged so safe steps remain safe.

  If both steps are i-steps then we have

  $$\begin{array}{lll} \sigma = \sigma[\forall\vec{\delta}\,\varrho[\forall\vec{\zeta}\beta\,\upsilon]_s\{\alpha \mapsto \xi_\alpha\}]_p & \overset{p}{\Longrightarrow}_i^- & \sigma[\forall\vec{\delta}\alpha\,\varrho[\forall\vec{\zeta}\beta\,\upsilon]_s]_p \\ & \overset{r}{\Longrightarrow}_i^+ & \sigma[\forall\vec{\delta}\alpha\,\varrho[\forall\vec{\zeta}\,\upsilon\{\beta \mapsto \xi_\beta\}]_s]_p = \tau \end{array}$$

  We know that $\beta \notin FV(\xi_\alpha)$, but α may occur in ξ_β. Assigning $\bar{\varrho} = \varrho\{\alpha \mapsto \xi_\alpha\}$ and $\bar{\upsilon} = \upsilon\{\alpha \mapsto \xi_\alpha\}$, we can swap these steps in the following way:

  $$\begin{array}{lll} \sigma = \sigma[\forall\vec{\delta}\,\bar{\varrho}[\forall\vec{\zeta}\beta\,\bar{\upsilon}]_s]_p & \overset{r}{\Longrightarrow}_i^+ & \sigma[\forall\vec{\delta}\,\bar{\varrho}[\forall\vec{\zeta}\,\bar{\upsilon}\{\beta \mapsto \xi_\beta\{\alpha \mapsto \xi_\alpha\}\}]_s]_p \\ & = & \sigma[\forall\vec{\delta}\,\varrho[\forall\vec{\zeta}\,\upsilon\{\beta \mapsto \xi_\beta\}]_s\{\alpha \mapsto \xi_\alpha\}]_p \\ & \overset{p}{\Longrightarrow}_i^- & \sigma[\forall\vec{\delta}\alpha\,\varrho[\forall\vec{\zeta}\,\upsilon\{\beta \mapsto \xi_\beta\}]_s]_p = \tau \end{array}$$

  the middle equality being a simple property of substitution. If the negative step was safe, it will remain safe since the type ξ_α does not change. If the positive step was safe, then $\xi_\beta = \bot$ or $\xi_\beta = \gamma$ for a variable γ bound at a positive position; in both cases $\alpha \notin FV(\xi_\beta)$, so $\xi_\beta\{\alpha \mapsto \xi_\alpha\} = \xi_\beta$;
- $p > r$. This situation is dual to the previous one. Given the derivation $\sigma \overset{p}{\Longrightarrow}^- \rho \overset{r}{\Longrightarrow}^+ \tau$ it suffices to consider

  $$\tau \to \bot \overset{0r}{\Longrightarrow}^- \rho \to \bot \overset{0p}{\Longrightarrow}^+ \sigma \to \bot$$

  where of course $0r < 0p$. ■

Lemma 5.4 *Given two types σ and τ such that $\sigma \sqsubseteq \tau$, there exists a normal derivation $\sigma \Longrightarrow\!\!\!\!\!\rightarrow^+ \Longrightarrow\!\!\!\!\!\rightarrow^- \tau$.*

Proof. Since by Lemma 4.1 there exists a derivation $\sigma \Longrightarrow\!\!\!\!\!\rightarrow \tau$, it suffices to use Lemma 5.3 appropriate numer of times in order to obtain a normal derivation. ■

After proving the above very important fact we can define the *weight* of an inequality.

Definition 5.5 `(weight of inequality)`
Weight of an inequality $\sigma \sqsubseteq \tau$ is the minimal weight of its normal derivations $\sigma \Longrightarrow\!\!\!\!\!\rightarrow^{+} \rho \Longrightarrow\!\!\!\!\!\rightarrow^{-} \tau$.

Now we can formally explain the meaning of the safe steps

Proposition 5.6 *Let $\sigma_1 \sqsubseteq \tau$ and let $\sigma_0 \Longrightarrow \sigma_1$ be a safe step. Then the weight of the inequality $\sigma_0 \sqsubseteq \tau$ is no greater than the weight of the inequality $\sigma_1 \sqsubseteq \tau$.*

Proof. This proposition is a straightforward consequence of Lemma 5.3. ■

Let us give an example of a rewriting step of weight 0, which is not safe. Consider the following one-step derivation of weight $weight(\xi)$:

$$\forall\alpha\,(\forall\beta\,(\bot \to \beta) \to \alpha) \overset{\epsilon}{\Longrightarrow}_i^{+} \forall\beta\,(\bot \to \beta) \to \xi$$

If we precede it by the zero-weight step

$$\forall\alpha\,((\bot \to \alpha) \to \alpha) = \forall\alpha\,((\bot \to \beta)\{\beta \mapsto \alpha\} \to \alpha) \overset{0}{\Longrightarrow}_i^{-} \forall\alpha\,(\forall\beta\,(\bot \to \beta) \to \alpha)$$

we get the derivation which is not normal. The first step is not safe, because the variable α is bound at a positive position ϵ and not negative, like the variable β. After the permutation of these steps, according to Lemma 5.3, we get the derivation

$$\forall\alpha\,((\bot \to \alpha) \to \alpha) \overset{\epsilon}{\Longrightarrow}_i^{+} ((\bot \to \xi) \to \xi) \overset{0}{\Longrightarrow}_i^{-} \forall\beta\,(\bot \to \beta) \to \xi$$

Its weight is equal $3 \times weight(\xi)$, that is more than the weight of the primary positive step, if only $weight(\xi) > 0$.

6 Bicoercible types

Types σ and τ are called *bicoercible* if $\sigma \sqsubseteq \tau$ and $\tau \sqsubseteq \sigma$. We show in the present section, that the bicoercibility relation may be identified with the least congruence containing the axioms

(B1) $\forall\alpha\forall\beta\,\sigma \sim \forall\beta\forall\alpha\,\sigma$

(B2) $\forall\alpha\,\sigma \sim \sigma\{\alpha \mapsto \bot\}$ (if all occurences of α in σ are positive)

The equivalence relation induced by the original Mitchell system was characterized by Tiuryn in [Tiu95]. It is defined by axioms **(B1)**, **(B2)** and the axiom

[1] A permutation the other way is not always possible. One cannot for example change the order of the steps $\forall\alpha\,\alpha \overset{\epsilon}{\Longrightarrow}_i^{+} (\beta \to \beta) \to \beta \overset{0}{\Longrightarrow}_i^{-} (\forall\alpha\,\alpha) \to \beta$

(B3) $\forall\alpha\,(\sigma_1 \to \sigma_2) \sim_{\mathrm{M}} \sigma_1 \to \forall\alpha\,\sigma_2 \quad$ (jeűli $\alpha \notin FV(\sigma_1)$)

which is not true in our system.[2] The rest of the present section is devoted to the proof that a relation $\sim$ defined as the least congruence containing axioms **(B1)** and **(B2)** is the equivalence relation induced by the quasi-order $\sqsubseteq$. To prove this we introduce the notion of reversible steps.

Definition 6.1 `(reversible step)`
A single rewriting step $\sigma \Longrightarrow \tau$ is called reversible, *if $\tau \sqsubseteq \sigma$.*

Lemma 6.2 *Let $\sigma \Longrightarrow_i^+ \tau$ and let p and r be such that $\sigma|_{pr} = \forall\vec{\delta}\,\alpha$, where α is bound at a positive position p. If the position pr in τ is not an occurence of a variable bound at a positive position, then the step $\sigma \Longrightarrow_i^+ \tau$ is not reversible.*

Proof. Suppose the step $\sigma \Longrightarrow_i^+ \tau$ is reversible. Then there exists a normal derivation $\tau \Longrightarrow\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\Rightarrow^- \sigma$.

If $\tau|_{pr} = \forall\vec{\zeta}\,\beta$, where β is free or bound at a negative position, then $\rho|_{pr} = \forall\vec{\vartheta}\,\beta$ as well, because during the positive phase of a derivation negative variables are not instantiated, and free variables cannot be instantiated at all.

If $\tau|_{pr} = \forall\vec{\zeta}\,(\xi_1 \to \xi_2)$, for some ξ_1, ξ_2, then similarily $\rho|_{pr} = \forall\vec{\vartheta}\,(\xi_1' \to \xi_2')$ (for some ξ_1', ξ_2'), because during the positive phase the weight of types does not decrease.

In both cases $\rho|_{pr}$ is not a variable bound at a positive position. It follows that neither is $\sigma|_{pr}$, because negative steps $\Longrightarrow_q^-$ do not change variable bindings, and the steps $\Longrightarrow_i^-$ do not create new variables bound at positive positions, a contradiction. ∎

The next lemma uses the following notation. Let σ be a type such that a position s is an occurence of a variable α bound by a quantifier at a position p (of course $p \leq s$). We denote by $\Pi_\sigma(s)$ the set of positions of all occurences of the variable α (bound by the quantifier $\forall\alpha$ at position p) in the type σ. If the type is clear from the context, the index σ will be omitted.

Lemma 6.3 *Let $\sigma \Longrightarrow\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\Rightarrow^- \sigma$ be a normal derivation of the inequality $\sigma \sqsubseteq \sigma$. Let p be a positive position and r such that pr is a negative position and $\sigma|_{pr} = \forall\vec{\vartheta}\,\alpha$, where α is a variable bound at a positive position p. Then in every type in the derivation $\sigma \Longrightarrow\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\Rightarrow^- \sigma$, the position pr is an occurence of a variable bound by a quantifier at the position p and the set $\Pi(pr)$ does not change.*

Proof. All steps in the considered derivation $\sigma \Longrightarrow\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\Rightarrow^- \sigma$ are of course reversible. The shape of type and the sign of variable bindings cannot be changed by positive steps, according to Lemma 6.2. By a similar argument neither can they be changed by negative steps, so they remain constant during the entire derivation.

Consider a the possible influence of a single positive step $\overset{s}{\Longrightarrow}_i^+$, accepted by Lemma 6.2, on the binding position of the variable α and the set $\Pi(pr)$

[2] It is not true that $\forall\alpha\,(\beta \to (\alpha \to \beta)) \sim \beta \to \forall\alpha\,(\alpha \to \beta)$

- if $s \not\geq p$, the step does not have any influence on the variable α;
- if $s \leq p$ and the quantifier $\forall\alpha$ at position p lies within the range of the instantiated quantifier, then the instantiation has no influence on α;
- if $s = p$ and $\forall\alpha$ is the instantiated quantifier, then the binding position can be changed into some positive p' such that $p = p'r'$ for some r'

$$\sigma_i[\forall\vec{\delta}\,\beta\,\varrho[\forall\vec{\zeta}\,\alpha\,\upsilon]_{r'}]_{p'} \overset{p}{\Longrightarrow}_i^+ \sigma_i[\forall\vec{\delta}\,\beta\,\varrho[\forall\vec{\zeta}\upsilon\{\alpha \mapsto \forall\vec{\vartheta}\,\beta\}]_{r'}]_{p'} = \sigma_{i+1}$$

- if $s \geq p$ and the instantiated quantifier lies within the range of $\forall\alpha$, then the set $\Pi(pr)$ can increase (we assume $s = pr''$ for some r'')

$$\sigma_i[\forall\vec{\zeta}\,\alpha\,\upsilon[\forall\vec{\delta}\gamma\,\chi]_{r''}]_p \overset{s}{\Longrightarrow}_i^+ \sigma_i[\forall\vec{\zeta}\,\alpha\,\upsilon[\forall\vec{\delta}\,\chi\{\gamma \mapsto \forall\vec{\vartheta}\,\alpha\}]_{r''}]_p = \sigma_{i+1}$$

According to this, the set $\Pi(pr)$ cannot be decreased nor can the binding position be turned into a bigger one in the prefix order. Since negative steps do not influence variables bound at positive positions, the binding position of α and the set $\Pi(pr)$ remain unchanged. ■

Lemma 6.4 *If $\sigma \Longrightarrow \tau$ is a reversible step then the statement $\sigma \sim \tau$ can be derived from the axioms* **(B1)** *and* **(B2)**.

Proof. If the step $\sigma \Longrightarrow \tau$ is a q-step then it suffices to use the **(B2)** axiom. Otherwise it is an i-step. Assume it is a positive step

$$\sigma = \sigma[\forall\vec{\delta}\,\alpha\,\varrho]_p \overset{p}{\Longrightarrow}_i^+ \sigma[\forall\vec{\delta}\varrho\{\alpha \mapsto \xi\}]_p = \tau$$

If $\alpha \notin FV(\varrho)$, we can use the **(B2)** axiom as above.

Let us suppose that $\alpha \in FV(\varrho)$. By Lemma 6.2 we obtain that $\xi = \forall\alpha_1 \ldots \alpha_n\,\beta$, where β is either one of the variables $\alpha_1, \ldots, \alpha_n$, or another variable bound at a positive position.

If $\beta = \alpha_i$ for some i, then by Lemma 6.2 the variable α may have only positive occurences in ϱ. In this case we can use the axiom **(B2)** to put $\bot$ in all occurences, and then add missing quantifiers, by the appropriate number of **(B2)** applications.

Otherwise β is another variable bound at a positive position p', where $p = p'r'$ for some r'. By Lemma 6.3 either $p' = p$ and $\beta \notin FV(\varrho)$ or all occurences of α and β are positive. In the first case we have to permute quantifiers at a position p by multiple application of the **(B1)** axiom and then eliminate the unnecessary $\forall\beta$ quantifier by **(B2)**, yielding a type that needs only a renaming of α to β.

In the second case we first replace all occurences of the variables α and β with $\bot$ by the **(B2)** axiom, and then by using the same axiom "the other way" we turn every inserted type $\bot$ into a new variable γ quantified by a new quantifier $\forall\gamma$ at a position p'. It remains to add missing empty quantifiers $\forall\alpha_1 \ldots \alpha_n$ where necessary and finally change the name of the bound variable γ back to β.

Since the same reasoning can be repeated for negative reversible $\sigma \Longrightarrow_i^- \tau$ step, we conclude that for every reversible step $\sigma \Longrightarrow \tau$, we can prove $\sigma \sim \tau$ using **(B1)** and **(B2)** axioms. ■

Lemma 6.5 *If the statement $\sigma \sim \tau$ can be proved by a single use of the axiom* **(B1)** *or* **(B2)** *then $\sigma \Longrightarrow\!\!\!\!\Rightarrow \tau$ and $\tau \Longrightarrow\!\!\!\!\Rightarrow \sigma$. Moreover these derivations use only safe steps.*

Proof. We have to consider four possibilities, depending on what axiom was used and what was the position sign. Since most of the reasoning is simple we will analyse only the case where the axiom **(B2)** was used at a positive position p

$$\sigma = \sigma[\forall\vec{\delta}\alpha\,\varrho]_p \sim \sigma[\forall\vec{\delta}\,\varrho\{\alpha \mapsto \bot\}]_p = \tau$$

The derivation from left to right can be done by a single i-step. Constructing a derivation from right to left is more complicated. Let $r_1, \ldots, r_n$ be all occurences of α in ϱ. For all i we have $\varrho\{\alpha \mapsto \bot\}|_{r_i} = \forall\vec{\zeta_i}\bot = \forall\vec{\zeta_i}\beta\,\beta$, so

$$\begin{array}{lll}
\sigma[\forall\vec{\delta}\,\varrho\{\alpha \mapsto \bot\}]_p & \stackrel{p}{\Longrightarrow}{}^{+}_{q} & \sigma[\forall\vec{\delta}\gamma\,\varrho\{\alpha \mapsto \bot\}]_p \\
& \stackrel{pr_1}{\Longrightarrow}{}^{+}_{i} & \sigma[\forall\vec{\delta}\gamma\,\varrho\{\alpha \mapsto \bot\}[\forall\vec{\zeta_1}\,\gamma]_{r_1}]_p \\
& \vdots & \\
& \stackrel{pr_n}{\Longrightarrow}{}^{+}_{i} & \sigma[\forall\vec{\delta}\gamma\,\varrho\{\alpha \mapsto \bot\}[\forall\vec{\zeta_1}\,\gamma]_{r_1} \cdots [\forall\vec{\zeta_n}\,\gamma]_{r_n}]_p \\
& = & \sigma[\forall\vec{\delta}\gamma\,\varrho\{\alpha \mapsto \gamma\}]_p \\
& \equiv & \sigma[\forall\vec{\delta}\alpha\,\varrho]_p
\end{array}$$

where γ is fresh. All steps $\stackrel{pr_i}{\Longrightarrow}{}^{+}_{i}$ are correct and safe because every r_i is positive, and the inserted variable γ is bound at a positive position p.

The other cases are left to the reader. ■

Theorem 1 `(characterization of bicoercibility)`
Inequalities $\sigma \sqsubseteq \tau$ and $\tau \sqsubseteq \sigma$ are both true if and only if $\sigma \sim \tau$.

Proof. If both inequalities $\sigma \sqsubseteq \tau$ and $\tau \sqsubseteq \sigma$ are true, then there exists a normal derivation $\sigma \Longrightarrow\!\!\!\!\Rightarrow^{+} \Longrightarrow\!\!\!\!\Rightarrow^{-} \tau$ containing only reversible steps. By Lemma 6.4 we have $\sigma \sim \tau$.

The opposite implication is also elementary. If $\sigma \sim \tau$, then by Lemma 6.5 and transitivity of $\sqsubseteq$ we obtain $\sigma \sqsubseteq \tau$ and $\tau \sqsubseteq \sigma$. ■

The property of bicoercible types, that will be used in consecutive sections is stated by the next proposition.

Proposition 6.6 *If $\sigma_0 \sim \sigma_1$ and $\sigma_1 \sqsubseteq \tau$ then the inequality $\sigma_0 \sqsubseteq \tau$ also holds and the weights of both inequalities are the same.*

Proof. This proposition is an easy consequence of Lemma 6.5 and Proposition 5.6. ■

7 Ordered derivations

In order to show the properties of $\sqsubseteq$ which will be directly used in the undecidability proof, we will use the ordered version of normal derivations. Intuitively,

positive rewriting steps will be ordred by their positions compared with the lexicographic ordering (induced by $0 \prec 1$). At one position, all q-steps will precede i-steps ordered from outermost to innermost. Negative steps will be ordered dually. The rewriting will then take place first at positive postion from the root towards the leaves, and then at negative positions, from the leaves upwards.

Definition 7.1 `(ordered derivations)`
A sequence of positive steps $\sigma_0 \stackrel{p_1}{\Longrightarrow}^+ \cdots \stackrel{p_n}{\Longrightarrow}^+ \sigma_n$ *is* ordered *if the following conditions are satisfied:*

1. *for all* i, j *if* $i < j$ *then* $p_i \preceq p_j$;
2. *for any given* p, $\stackrel{p}{\Longrightarrow}_q^+$ *steps precede* $\stackrel{p}{\Longrightarrow}_i^+$ *steps;*
3. *there are no pairs of steps*

$$\begin{array}{lll} \sigma_j[\forall\vec{\delta}\,\alpha\,\vec{\zeta}\beta\;\varrho]_p & \stackrel{p}{\Longrightarrow}_i^+ & \sigma_j[\forall\vec{\delta}\,\alpha\,\vec{\zeta}\;\varrho\{\beta\mapsto\xi_\beta\}]_p \\ & \stackrel{p}{\Longrightarrow}_i^+ & \sigma_j[\forall\vec{\delta}\,\vec{\zeta}\;\varrho\{\beta\mapsto\xi_\beta\}\{\alpha\mapsto\xi_\alpha\}]_p \end{array}$$

A sequence of negative steps $\tau_0 \Longrightarrow^- \cdots \Longrightarrow^- \tau_n$ *is* ordered*, if the dual sequence of positive steps* $\tau_n \to \gamma \Longrightarrow^+ \cdots \Longrightarrow^+ \tau_0 \to \gamma$ *(for* γ *fresh) is ordered.*

A normal derivation $\sigma \Longrightarrow\!\!\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\!\!\Rightarrow^- \tau$ *is called* ordered *if both sequences* $\sigma \Longrightarrow\!\!\!\!\!\Rightarrow^+ \rho$ *and* $\rho \Longrightarrow\!\!\!\!\!\Rightarrow^- \tau$ *are ordered.*

We will prove now that all derivations may be ordered. We need the following key lemma.

Lemma 7.2 *If a two-step sequence* $\sigma \stackrel{p}{\Longrightarrow}^+ \rho \stackrel{p'}{\Longrightarrow}^+ \tau$ *is not ordered, then there exists a type* ρ' *such that the derivation* $\sigma \Longrightarrow^+ \rho' \Longrightarrow^+ \tau$ *is ordered.*

Proof. In most cases we can easily permute rewriting steps. The only difficulty arises when we have two i-steps and p' is a prefix of p. Let r be such that $pr = p'$ (possibly $r = \epsilon$). We have

$$\begin{array}{lcll} \sigma & = & \sigma[\forall\vec{\delta}\alpha\,\varrho[\forall\vec{\zeta}\beta\,\upsilon]_r]_{p'} & \stackrel{p}{\Longrightarrow}_i^+ \; \sigma[\forall\vec{\delta}\alpha\,\varrho[\forall\vec{\zeta}\,\upsilon\{\beta\mapsto\xi_\beta\}]_r]_{p'} \\ & & & \stackrel{p'}{\Longrightarrow}_i^+ \; \sigma[\forall\vec{\delta}\,\varrho[\forall\vec{\zeta}\,\upsilon\{\beta\mapsto\xi_\beta\}]_r\{\alpha\mapsto\xi_\alpha\}]_{p'} \;=\; \tau \end{array}$$

Using the fact that $\upsilon\{\beta\mapsto\xi_\beta\}\{\alpha\mapsto\xi_\alpha\} = \upsilon\{\alpha\mapsto\xi_\alpha\}\{\beta\mapsto\xi_\beta\{\alpha\mapsto\xi_\alpha\}\}$ and assigning $\overline{\varrho} = \varrho\{\alpha\mapsto\xi_\alpha\}$ and $\overline{\upsilon} = \upsilon\{\alpha\mapsto\xi_\alpha\}$ we can do the following

$$\begin{array}{lcll} \sigma & = & \sigma[\forall\vec{\delta}\alpha\,\varrho[\forall\vec{\zeta}\beta\,\upsilon]_r]_{p'} & \stackrel{p}{\Longrightarrow}_i^+ \; \sigma[\forall\vec{\delta}\,\varrho[\forall\vec{\zeta}\beta\,\upsilon]_r\{\alpha\mapsto\xi_\alpha\}]_{p'} \\ & & & = \quad \sigma[\forall\vec{\delta}\,\overline{\varrho}[\forall\vec{\zeta}\beta\,\overline{\upsilon}]_r]_{p'} \\ & & & \stackrel{p'}{\Longrightarrow}_i^+ \; \sigma[\forall\vec{\delta}\,\overline{\varrho}[\forall\vec{\zeta}\,\overline{\upsilon}\{\beta\mapsto\xi_\beta\{\alpha\mapsto\xi_\alpha\}\}]_r]_{p'} \\ & & & = \quad \sigma[\forall\vec{\delta}\,\varrho[\forall\vec{\zeta}\,\upsilon\{\beta\mapsto\xi_\beta\}]_r\{\alpha\mapsto\xi_\alpha\}]_{p'} \;=\; \tau \end{array}$$

■

Lemma 7.3 (ordering derivations)
If $\sigma \sqsubseteq \tau$, then there exists an ordered derivation $\sigma \Longrightarrow^{+} \Longrightarrow^{-} \tau$, of weight equal to the weight of the inequality.

Proof. According to Lemma 7.2, we conclude that any sequence of positive steps may be ordered. Using the dual method, we can order the negative steps as well.

Let $\sigma \Longrightarrow^{+} \rho \Longrightarrow^{-} \tau$ be a normal derivation of the inequality $\sigma \sqsubseteq \tau$ of minimal weight. Ordering of positive and negative steps do not change the type ρ, so the weight of the derivation $\sigma \Longrightarrow^{+} \rho \Longrightarrow^{-} \tau$ equal $2 \times \mathit{weight}(\rho) - \mathit{weight}(\sigma) - \mathit{weight}(\tau)$ does not change. ■

8 Decomposing inequalities

The present section shows the properties of $\sqsubseteq$ that will be directly used in the proof of undecidability.

Lemma 8.1 *Let $\sigma \sqsubseteq \tau$ and let $\sigma|_p = \varrho_\sigma$ and $\tau|_p = \varrho_\tau$ be subterms such that no variable that is free in ϱ_σ (resp. in ϱ_τ) is bound in σ (resp. τ). If p is positive, then $\varrho_\sigma \sqsubseteq \varrho_\tau$ and if p is negative then $\varrho_\tau \sqsubseteq \varrho_\sigma$.*

Proof. Let $\sigma = \sigma_0 \stackrel{p_1}{\Longrightarrow}{}^{+} \cdots \stackrel{p_n}{\Longrightarrow}{}^{+} \sigma_n = \tau_m \stackrel{q_m}{\Longrightarrow}{}^{-} \cdots \stackrel{q_1}{\Longrightarrow}{}^{-} \tau_0 = \tau$ be an orderd derivation of the inequality $\sigma \sqsubseteq \tau$. Suppose p is positive. Since the sequences $p_1, \ldots, p_n$ and $q_1, \ldots, q_m$ are lexicographically ordered, each phase of the derivation can be divided into three parts. Consider the positive phase.

First there come steps at positions $p_1, \ldots, p_{k-1}$ smaller then p. They do not influence the subterm at a positon p since no free variable of ϱ_σ is bound above. Then there come all steps at positions $p_k, \ldots, p_{k'}$ which have p as a prefix. They can be seen as a derivation issued from ϱ_σ. The only problem arises when some variable in an inserted type gets bound by a quantifier above p. But since this quantifier cannot be instantiated (the derivation is ordered) this variable has to be removed by some negative steps. Hence we can replace a bound variable by a free one, because in any case it will be removed.

The last sequence of steps, at positions $p_{k'+1}, \ldots, p_n$, has no influence on the subtype at position p. An analogous partition can be done with the negative phase, so the middle parts of each phase form a desired derivation $\varrho_\sigma \Longrightarrow \varrho_\tau$ (or the other way if p is negative). ■

Lemma 8.2 *The inequality $\sigma' \to \sigma'' \sqsubseteq \tau' \to \tau''$ holds if and only if both inequalities $\sigma' \sqsupseteq \tau'$ and $\sigma'' \sqsubseteq \tau''$ hold. Moreover, if all inequalities are true then the weight of the first one is equal to the sum of weights of the others.*

Proof. The implication from left to right is an easy consequence of the rule **(→-context)** and from right to left — of the last lemma.

The second statement of the lemma is also easy. Having two normal derivations of minimal weight $\tau' = \tau'_0 \stackrel{t_1}{\Longrightarrow}{}^{+} \cdots \stackrel{t_k}{\Longrightarrow}{}^{+} \tau'_k = \sigma'_l \stackrel{u_l}{\Longrightarrow}{}^{-} \cdots \stackrel{u_1}{\Longrightarrow}{}^{-} \sigma'_0 = \sigma'$

and $\sigma'' = \sigma''_0 \stackrel{p_1}{\Longrightarrow}{}^+ \cdots \stackrel{p_n}{\Longrightarrow}{}^+ \sigma''_n = \tau''_m \stackrel{q_m}{\Longrightarrow}{}^- \cdots \stackrel{q_1}{\Longrightarrow}{}^- \tau''_0 = \tau''$ we can combine them (reversing the steps of the first one) into

$$\begin{array}{rl} \sigma'_0 \to \sigma''_0 \stackrel{0u_1}{\Longrightarrow}{}^+ \cdots \stackrel{0u_l}{\Longrightarrow}{}^+ \sigma'_l \to \sigma''_0 \stackrel{1p_1}{\Longrightarrow}{}^+ \cdots \stackrel{1p_n}{\Longrightarrow}{}^+ \sigma'_l \to \sigma''_n & = \\ = \quad \tau'_k \to \tau''_m \stackrel{1q_m}{\Longrightarrow}{}^- \cdots \stackrel{1q_1}{\Longrightarrow}{}^- \tau'_k \to \tau''_0 \stackrel{0t_k}{\Longrightarrow}{}^- \cdots \stackrel{0t_k}{\Longrightarrow}{}^- \tau'_0 \to \tau''_0 & \end{array} \tag{2}$$

So the weight of the inequality $\sigma' \to \sigma'' \sqsubseteq \tau' \to \tau''$ is no greater than the sum of weights of $\sigma' \sqsupseteq \tau'$ and $\sigma'' \sqsubseteq \tau''$. We will now show that it cannot be smaller either.

Let us take an ordered derivation of minimal weight of the inequality $\sigma' \to \sigma'' \sqsubseteq \tau' \to \tau''$. It starts with some $\stackrel{\epsilon}{\Longrightarrow}{}^+$ steps, but an easy analysis proves that those steps are meaningless. So the rest of the derivation consist of steps either at a position $0p'$ or $1r'$, just like in the derivation (2). So it can be split into the component derivations the weights of which sum to the weight of the primary inequality. ■

Lemma 8.3 *Let σ and τ be types. Assume that there exist two paths: positive p and negative r such that $\sigma|_p = \sigma|_r = \alpha$ and $\tau|_p = \tau|_r = \alpha$ for a free variable α. Suppose furthermore that τ is not of the form $\forall\gamma\,\tau'$. Then*

1. *$\forall\alpha\,\sigma \sqsubseteq \tau\{\alpha \mapsto \xi\}$ if and only if $\sigma\{\alpha \mapsto \xi\} \sqsubseteq \tau\{\alpha \mapsto \xi\}$;*
2. *if both inequalities hold then the weight of the latter is no smaller than the weight of the former;*
3. *moreover if $weight(\xi) > 0$ then the weight of the first inequality is strictly greater than the weight of the second.*

Proof. If the inequality $\sigma\{\alpha \mapsto \xi\} \sqsubseteq \tau\{\alpha \mapsto \xi\}$ holds, then since $\forall\alpha\,\sigma \stackrel{\epsilon}{\Longrightarrow}{}^i_+ \sigma\{\alpha \mapsto \xi\}$, we have the following wychodzi!

$$\forall\alpha\,\sigma \sqsubseteq \tau\{\alpha \mapsto \xi\} \tag{3}$$

In order to show the opposite implication suppose the latter inequality holds. Consider its ordered derivation of minimal weight. It begins with some meaningless $\stackrel{\epsilon}{\Longrightarrow}{}^+$ steps. All they can do is to add some empty quantifiers after the initial $\forall\alpha$. Now the latter must be instantiated:

$$\forall\alpha\,\sigma' \stackrel{\epsilon}{\Longrightarrow}{}^+_i \sigma'\{\alpha \mapsto \chi\} \tag{4}$$

where σ' stands for $\forall\beta_1 \ldots \beta_n\,\sigma$, and $\beta_1, \ldots, \beta_n$ are the quantifiers added by first $\stackrel{\epsilon}{\Longrightarrow}{}^+$ steps. Of course $\beta_1, \ldots, \beta_n \notin FV(\sigma'\{\alpha \mapsto \chi\})$. By Lemma 8.1, $\xi \sim \chi$, so also $\sigma\{\alpha \mapsto \xi\} \sim \sigma'\{\alpha \mapsto \chi\}$. This yields the inequality

$$\sigma\{\alpha \mapsto \xi\} \sqsubseteq \tau\{\alpha \mapsto \xi\}$$

Its weight, by Proposition 6.6 is no greater than the weight of (3) minus the weight of (4), which is strictly positive, when $weight(\xi)$ is so. Since $weight(\xi) = weight(\chi)$ we have the desired conclusion. ■

9 Encoding a machine

This section completes the proof of undecidability of the relation $\sqsubseteq$. After proving Lemmas 8.2 and 8.3, we know that $\sqsubseteq$ has all properties of the relation $\sqsubseteq_{\mathrm{M}}$ used in [TU95] to encode the halting problem of stack register machines. We show briefly the idea of this encoding.

A stack register machine $\mathcal{M}$ is a deterministinc computing device with two main registers V_1 and V_2 and a finite number of auxiliary registers $v_1, \ldots, v_n$. The registers can hold nonempty words over the finite alphabet Σ, containing instruction labels with one special end label e. A machine step consists of taking the top label from the first main register, executing the assigned instruction on the auxiliary registers and finally switching the main ones. If the read label is the last one in V_1, we copy the contents of the first auxiliary register, before instruction execution. Machine instructions are parallel assignments of one registers to another, with pushing some new labels at the top. For a given label $\ell \neq e$ we write

$$\ell(r_1, \ldots, r_n) = (w_1 r_{i_1}, \ldots, w_n r_{i_n}) \qquad \text{where } w_i \in \Sigma^*$$

Given an instantaneous description — a tuple $(V_1, V_2, v_1, \ldots, v_n)$ of $n+2$ words — a next step leads to the following configuration, depending on the contents of the first main register:

- if $V_1 = eV_1'$, the machine halts;
- if $V_1 = \ell V_1'$, where $\ell \neq e$ and $V_1 \neq \epsilon$, the next instantaneous description is $(V_2, V_1', v_1', \ldots, v_n')$, for $\ell(v_1, \ldots, v_n) = (v_1', \ldots, v_n')$;
- if $V_1 = \ell$, for $\ell \neq e$, the next configuration is $(V_2, v_1, v_1', \ldots, v_n')$, for $v_1', \ldots, v_n'$ as above.

The encoding idea is the following. For each label $\ell \in \Sigma$ we construct a type σ_ℓ hiding the associated instruction. Then we give the word composition rules yielding from a given configuration the $n+2$ types, $\sigma_{V_1}, \sigma_{V_2}, \sigma_{v_1}, \ldots, \sigma_{v_n}$, containing one special free variable γ. The key to the undecidability proof is the following proposition:

Proposition 9.1 *Given a stack register machine $\mathcal{M}$, it halts for an instantaneous description $(V_1, V_2, v_1, \ldots, v_n)$ if and only if the encoding inequality holds*

$$\sigma_{V_1} \sqsubseteq (\sigma_{v_1} \to \sigma_{v_1}) \to \cdots (\sigma_{v_n} \to \sigma_{v_n}) \to \sigma_{V_2} \to \gamma \tag{5}$$

Proof sketch:[3] The equivalence is based on the following properties of the encoding and the relation $\sqsubseteq$

- if $(V_1, V_2, v_1, \ldots, v_n)$ is an end configuration, i.e. $V_1 = eV_1'$, then $\sigma_{V_1} = \bot$;

[3] For the details concerning the encoding and the proof of the proposition we refer the reader to [TU95]

- if $V_1 = \ell V_1'$ for $V_1' \neq \epsilon$ and $\ell(r_1, \ldots, r_n) = (w_1 r_{i_1}, \ldots, w_n r_{i_n})$, then we have

$$\begin{aligned}\sigma_{V_1} \quad = \quad & \forall \alpha_1 \ldots \alpha_n \Big((\alpha_1 \to \alpha_1) \to \ldots \to (\alpha_n \to \alpha_n) \to \\ & \Big((\sigma_{w_1}[\alpha_{i_1}] \to \sigma_{w_1}[\alpha_{i_1}]) \to \cdots \to (\sigma_{w_n}[\alpha_{i_n}] \to \sigma_{w_n}[\alpha_{i_n}]) \to \\ & \sigma_{V_1'} \to \gamma \Big) \to \gamma \Big)\end{aligned}$$

  If the encoding inequality (5) holds, then by using Lemma 8.3 n times, we know it is equivalent to the inequality

$$\begin{aligned}& (\sigma_{v_1} \to \sigma_{v_1}) \to \ldots \to (\sigma_{v_n} \to \sigma_{v_n}) \to \\ & \Big((\sigma_{w_1}[\sigma_{v_{i_1}}] \to \sigma_{w_1}[\sigma_{v_{i_1}}]) \to \ldots \to (\sigma_{w_n}[\sigma_{v_{i_n}}] \to \sigma_{w_n}[\sigma_{v_{i_n}}]) \to \sigma_{V_1'} \to \gamma \Big) \to \gamma \\ \sqsubseteq \quad & (\sigma_{v_1} \to \sigma_{v_1}) \to \ldots \to (\sigma_{v_n} \to \sigma_{v_n}) \to \sigma_{V_2} \to \gamma\end{aligned}$$

  After application of Lemma 8.2 $(n+1)$ times, we obtain the equivalence with the encoding inequality after one step of the execution

$$\sigma_{V_2} \sqsubseteq (\sigma_{w_1}[\sigma_{v_{i_1}}] \to \sigma_{w_1}[\sigma_{v_{i_1}}]) \to \ldots \to (\sigma_{w_n}[\sigma_{v_{i_n}}] \to \sigma_{w_n}[\sigma_{v_{i_n}}]) \to \sigma_{V_1'} \to \gamma$$

  The weight of the latter inequality is no greater then the weight of 5), and the difference becomes strictly positive if at least one type $\sigma_{v_1}, \ldots, \sigma_{v_n}$ was of a non-zero weight;

- similarily, if $V_1 = \ell$, for ℓ as above, we have

$$\begin{aligned}\sigma_{V_1} \quad = \quad & \forall \alpha_1 \ldots \alpha_n \Big((\alpha_1 \to \alpha_1) \to \ldots \to (\alpha_n \to \alpha_n) \to \\ & \Big((\sigma_{w_1}[\alpha_{i_1}] \to \sigma_{w_1}[\alpha_{i_1}]) \to \cdots \to (\sigma_{w_n}[\alpha_{i_n}] \to \sigma_{w_n}[\alpha_{i_n}]) \to \\ & \alpha_1 \to \gamma \Big) \to \gamma \Big)\end{aligned}$$

  and by the analogous reasoning we obtain the equivalence of (5) with the inequality

$$\sigma_{V_2} \sqsubseteq (\sigma_{w_1}[\sigma_{v_{i_1}}] \to \sigma_{w_1}[\sigma_{v_{i_1}}]) \to \ldots \to (\sigma_{w_n}[\sigma_{v_{i_n}}] \to \sigma_{w_n}[\sigma_{v_{i_n}}]) \to \sigma_{v_1} \to \gamma$$

It is obvious that if the configuration $(V_1, V_2, v_1, \ldots, v_n)$ is an end one, the encoding inequality holds. Otherwise we have the equivalence

$$\begin{aligned}& \sigma_{V_1} \sqsubseteq (\sigma_{v_1} \to \sigma_{v_1}) \to \cdots (\sigma_{v_n} \to \sigma_{v_n}) \to \sigma_{V_2} \to \gamma \\ \iff \quad & \sigma_{U_1} \sqsubseteq (\sigma_{u_1} \to \sigma_{u_1}) \to \cdots (\sigma_{u_n} \to \sigma_{u_n}) \to \sigma_{U_2} \to \gamma\end{aligned}$$

where $(U_1, U_2, u_1, \ldots, u_n)$ is the machine configuration after one step of the execution. Moreover the weight of the latter inequality is no greater then the weight of the former and if at least one of $v_1, \ldots, v_n$ does not start with e, the difference of the weights is strictly positive.

Of course if the machine halts, the encoding inequality holds for all configurations leading to the end one.

The proof of the other implication is slightly more difficult. If the encoding inequality holds then with each step the weight of the encoding inequality decreases or (if all auxiliary registers begin with the e label) the number of labels in the main registers decreases while the weight of the encoding inequality remains unchanged. When the first main register becomes empty, its contents is replaced by the word beginning with the end label which provides termination of the execution in the next step. ■

Theorem 2
The relation $\sqsubseteq$ of polymorphic subtyping without distributivity is undecidable.

Proof. The result follows from the our Propsition 9.1 and Lemma 16 from the paper of Tiuryn and Urzyczyn [TU95] that reduces the undecidable halting problem of deterministic two-counter automata to the halting problem of stack register machines. ■

10 Conclusions

The same proof would also be valid if we removed yet another axiom from the definition of subtyping — the axiom of generalization **(quant)**. The only difference would emerge in Section 6, but Proposition 6.6 remains true which enables us to continue with the undecidability proof.

Another direction of searching for decidable subtyping relation is the predicative version of System **F**, presented by Daniel Leivant [Lei91]. Some work has already been done in this field (see [**?**]), but we believe that further investigation should lead to interesting results.

References

[Gir89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.

[Jim95] T. Jim, "System F plus subsumption reduces to Mitchell's subtyping relation", Manuscript (1995).

[Lei91] D. Leivant, "Finitely stratified polymorphism", *Information and Computation*, **93** (1), 1991, 93–113.

[LMS95] G. Longo, K. Milsted, and S. Soloviev, "A logic of subtyping", *Proc. 10-th IEEE Symp. Logic in Computer Science*, San Diego (1995), pp. 292-299.

[Mit88] J.C. Mitchell, "Polymorphic type inference and containment", *Information and Computation*, **76** (2-3), 1988, pp. 211–249.

[Tiu95] J. Tiuryn, "Equational Axiomatization of Bicoercibility for Polymorphic Types", *Proc. Conf. Foundations of Software Technology and Teoretical Computer Science'95*, Bangalore, India; 18-20 Dec 1995. Lecture Notes in Computer Science 1026, pp. 166-179.

[TU95] J. Tiuryn and P. Urzyczyn, "The Subtyping Problem For Second-Order Types in Undecidable" *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science*, pp. 74-85, 27-30 July 1996.

[Wel94] J.B. Wells, "Typability and type checking in the second-order λ-calculus calculus are equivalent and undecidable", *Proc. 9th LICS*, IEEE, 1994, pp. 176–185.

[Wel95] J.B. Wells, "The Undecidabilility of Mitchell's Subtyping Relationship", Technical Report, Computer Science Department, Boston University, Number 95-019, December 10 1995.

[Wel96] J.B. Wells, "Typability is Undecidable for F+Eta" Technical Report, Computer Science Department, Boston University, Number 96-022, March 9 1996.