Polymorphic Subtyping Without Distributivity

Jacek Chrząszcz*

Institute of Informatics,
Warsaw University,
ul. Banacha 2, 02-097 Warsaw, Poland.
email: chrzaszcz@mimuw.edu.pl

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 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: 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 is characterized by $\forall\alpha_1 \ldots \alpha_n\, \sigma \;\; \sqsubseteq_{\mathbf{F}} \;\; \forall\beta_1 \ldots \beta_m\, \sigma\{\alpha_1 \mapsto \rho_1, \ldots, \alpha_n \mapsto \rho_n\}$ 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 [7] 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: $\sigma_1 \to \tau_1 \sqsubseteq_{\mathrm{M}} \sigma_2 \to \tau_2$ if $\sigma_2 \sqsubseteq_{\mathrm{M}} \sigma_1$ and $\tau_1 \sqsubseteq_{\mathrm{M}} \tau_2$. 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 **F**. Unfortunately the resulting system $\mathbf{F}_\eta$ has all the bad undecidability properties of system **F**. Typability and type-checking are

* The author is partly supported by Polish KBN Grant 8 T11C 034 10.

undecidable in both systems (see [10] for **F** and [12] for typability and [9] for type-checking in $\mathbf{F}_\eta$) and so is Mitchell subtyping relation [9] (see also [11] 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 [9] 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 [6], which deeply incorporates the distributivity axiom.

Another approach is taken by Wells [11], 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 $\boldsymbol{\gamma}$ are all variables in $\sigma_1, \tau_1,\ \sigma_2, \tau_2$ then $\forall\beta\,(\Pi(\beta,\beta,\beta,\beta) \to \bot) \ \sqsubseteq_{\mathrm{M}}\ (\forall\boldsymbol{\gamma}\,\Pi(\sigma_1,\tau_1,\sigma_2,\tau_2)) \to \bot$ 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 undecidability result is based on a reduction similar to that in the work of Tiuryn and Urzyczyn. In fact their encoding depends only on the notion of weight of an inequality and on two properties of the subtyping relation $\sqsubseteq_{\mathrm{M}}$ (Lemmas 12 and 14). The present paper, instead of basing the notion of weight on Longo-Milsted-Soloviev derivations, defines a type rewriting system which is a straightforward translation of original Mitchell axioms. The notion of weight is defined using this system, and its further analysis enables us to show that the subtyping without distributivity has all the properties of $\sqsubseteq_{\mathrm{M}}$ (see Lemmas 10 and 11) on which depends the encoding from [9].

The organization of the paper is as follows. After the preliminaries, the rewriting and its correspondence with the subtyping axiomatization 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 [8] for 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 phase. It is somewhat similar to Wells's syntax-driven system [11], 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 8). Finally, in Section 9 we conclude the paper with some suggestions 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 grammar $\mathcal{T} ::= V \mid \forall V\, \mathcal{T} \mid \mathcal{T} \to \mathcal{T}$.

Because of quantifiers, variables occurring 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 occurring in it is denoted by $FV(\sigma)$. 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.

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

The number of arrows appearing in a given type is called its *weight*.

Substitutions are denoted $\{\alpha_1 \mapsto \xi_1, \ldots, \alpha_n \mapsto \xi_n\}$ and we use postfix notation for their application to types. Substitutions bind stronger than quantifiers, so in the type $\forall\beta\,\sigma\{\alpha \mapsto \xi\}$ the free occurrences of β in the type ξ will be bound by the quantifier $\forall\beta$.

Polymorphic 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 strings 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 string is denoted by ϵ, and given two strings p, r their concatenation is written as pr.

Positions will 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:

$$(\boldsymbol{\to}\textbf{-context})\ \frac{\sigma_2 \sqsubseteq \sigma_1 \quad \tau_1 \sqsubseteq \tau_2}{\sigma_1 \to \tau_1 \sqsubseteq \sigma_2 \to \tau_2}$$

$$(\boldsymbol{\forall}\textbf{-context})\ \frac{\sigma \sqsubseteq \tau}{\forall\alpha\,\sigma \sqsubseteq \forall\alpha\,\tau}$$

$$(\textbf{trans})\ \frac{\sigma \sqsubseteq \rho \quad \rho \sqsubseteq \tau}{\sigma \sqsubseteq \tau}$$

Our definition differs from the original one introduced by Mitchell in [7] in the lack of the distributivity axiom:

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

4 Type rewriting

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

$$C^+[\forall\alpha\,\sigma]_p \overset{p}{\Longrightarrow}{}_i^+ C^+[\sigma\{\alpha \mapsto \rho\}]_p \qquad C^-[\sigma\{\alpha \mapsto \rho\}]_r \overset{r}{\Longrightarrow}{}_i^- C^-[\forall\alpha\,\sigma]_r$$

$$C^+[\sigma]_p \overset{p}{\Longrightarrow}{}_q^+ C^+[\forall\alpha\,\sigma]_p \quad \text{if } \alpha \notin FV(\sigma) \qquad C^-[\forall\alpha\,\sigma]_r \overset{r}{\Longrightarrow}{}_q^- C^-[\sigma]_r \quad \text{if } \alpha \notin FV(\sigma)$$

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 $\Longrightarrow\!\!\!\!\!\!\!\Longrightarrow$. Given a derivation step $\sigma \Longrightarrow \tau$, 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.

Lemma 1. *For any types σ and τ we have $\sigma \sqsubseteq \tau$ if and only if $\sigma \Longrightarrow\!\!\!\!\!\!\!\Longrightarrow \tau$.*

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 1 (normal derivation).
A derivation $\sigma \Longrightarrow\!\!\!\!\!\!\!\Longrightarrow \tau$ is called normal*, if it is of the form $\sigma \Longrightarrow\!\!\!\!\!\!\!\Longrightarrow^+ \rho \Longrightarrow\!\!\!\!\!\!\!\Longrightarrow^- \tau$ 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 showing how to normalize derivations, define the notion of safe steps.

Definition 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 instantiation 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 Definition 2 expresses the following situations:

$$C^+[\forall\beta\, C^+[\forall\alpha\,\sigma]] \Longrightarrow_i^+ C^+[\forall\beta\, C^+[\sigma\{\alpha \mapsto \beta\}]]$$

$$C^-[\forall\beta\, C^+[\sigma\{\alpha \mapsto \beta\}]] \Longrightarrow_i^- C^-[\forall\beta\, C^+[\forall\alpha\,\sigma]]$$

All safe steps have the weight 0. In addition, a safe step can be inserted in a normal 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 normalize a derivation.

Lemma 2. *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 similarly if $\rho \Longrightarrow^+ \tau$ is a safe step then so will be $\sigma \Longrightarrow^+ \rho'$.*

Proof. Let $\sigma \overset{p}{\Longrightarrow}^- \rho \overset{r}{\Longrightarrow}^+ \tau$. Depending on p and r we have three possibilities to consider: $p \not\gtrless r$, $p < r$, and $p > r$. The first one is easy, the other two, dual to each other, are also easy when any of the rewriting steps is a q-step. Otherwise both steps are instantiations and we use the fact that $\upsilon\{\beta \mapsto \xi_\beta\{\alpha \mapsto \xi_\alpha\}\} = \upsilon\{\beta \mapsto \xi_\beta\}\{\alpha \mapsto \xi_\alpha\}$ for any types υ, ξ_α, and ξ_β. □

Lemma 3. *Given two types σ and τ such that $\sigma \sqsubseteq \tau$, there exists a normal derivation $\sigma \Longrightarrow\!\!\!\!\!\Rightarrow^+ \Longrightarrow\!\!\!\!\!\Rightarrow^- \tau$.*

Now we define the *weight* of an inequality and explain the meaning of safe steps.

Definition 3 (weight of inequality).
The weight *of an inequality $\sigma \sqsubseteq \tau$ is the minimal weight of a normal derivation $\sigma \Longrightarrow\!\!\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\!\!\Rightarrow^- \tau$.*

Proposition 4. *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 $\sigma_1 \sqsubseteq \tau$.*

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

Let us give an example of a rewriting step of weight 0, which is not safe.

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

It is not safe, because the variable α is bound at a positive position, unlike the variable β. If we put this step in front of a one-step derivation of weight $\mathit{weight}(\xi)$:

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

and permute the steps according to Lemma 2, we get a normal derivation of weight $3 \times \mathit{weight}(\xi)$:

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

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) $\quad \forall\alpha\forall\beta\,\sigma \sim \forall\beta\forall\alpha\,\sigma$

(B2) $\quad \forall\alpha\,\sigma \sim \sigma\{\alpha \mapsto \bot\}$ (if all occurrences of α in σ are positive[2])

The equivalence relation induced by the original Mitchell system was characterized by Tiuryn in [8]. It is defined by axioms **(B1)**, **(B2)** and the axiom

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

which is not true in our system.[3] The rest of the present section is devoted to the proof that the relation $\sim$, defined as the least congruence containing axioms **(B1)** and **(B2)**, is the equivalence relation induced by the quasi-order $\sqsubseteq$.

Lemma 5. *If $\sigma \Longrightarrow \tau$ and $\tau \sqsubseteq \sigma$ then the statement $\sigma \sim \tau$ can be derived from the axioms* **(B1)** *and* **(B2)***.*

Proof. Since both situations are similar, assume $\sigma \Longrightarrow \tau$ is a positive step. By analyzing what changes introduced by the first step can be undone by further steps of a normal derivation $\sigma \Longrightarrow^{+} \tau \Longrightarrow\!\!\!\!\!\Rightarrow^{+} \rho \Longrightarrow\!\!\!\!\!\Rightarrow^{-} \sigma$, we conclude that inserted types can only be variables, that the sign of a variable binding must not change, and finally that when a variable bound at a positive position has negative occurrences (or vice versa) the changes to its binding can only be trivial. If follows that all kinds of acceptable changes can be simulated by **(B1)** and **(B2)**. □

Lemma 6. *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.*

[2] It covers also the situation $\alpha \notin FV(\sigma)$

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

Proof. There are four possibilities, depending on what axiom was used and what was the position sign. Since most of the reasoning is simple we analyze only the use of **(B2)** at a positive position: $\sigma = \sigma[\forall\boldsymbol{\delta}\alpha\,\varrho]_p \sim \sigma[\forall\boldsymbol{\delta}\,\varrho\{\alpha\mapsto\bot\}]_p = \tau$.

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 occurrences of α in ϱ. For all i we have $\varrho\{\alpha\mapsto\bot\}|_{r_i} = \forall\boldsymbol{\zeta_i}\bot = \forall\boldsymbol{\zeta_i}\beta\,\beta$, so

$$\begin{aligned}
\sigma[\forall\boldsymbol{\delta}\,\varrho\{\alpha\mapsto\bot\}]_p \;&\xRightarrow{p}{}^{+}_{q}\; \sigma[\forall\boldsymbol{\delta}\gamma\,\varrho\{\alpha\mapsto\bot\}]_p\\
&\xRightarrow{pr_1}{}^{+}_{i}\; \sigma[\forall\boldsymbol{\delta}\gamma\,\varrho\{\alpha\mapsto\bot\}[\forall\boldsymbol{\zeta_1}\,\gamma]_{r_1}]_p \;\xRightarrow{pr_2}{}^{+}_{i}\;\cdots\\
\cdots\;&\xRightarrow{pr_n}{}^{+}_{i}\; \sigma[\forall\boldsymbol{\delta}\gamma\,\varrho\{\alpha\mapsto\bot\}[\forall\boldsymbol{\zeta_1}\,\gamma]_{r_1}\cdots[\forall\boldsymbol{\zeta_n}\,\gamma]_{r_n}]_p\\
&=\; \sigma[\forall\boldsymbol{\delta}\gamma\,\varrho\{\alpha\mapsto\gamma\}]_p \;=\; \sigma[\forall\boldsymbol{\delta}\alpha\,\varrho]_p
\end{aligned}$$

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

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

For the rest of the proof the most important is the following proposition which follows from Lemma 6 and Proposition 4.

Proposition 7. *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.*

7 Decomposing inequalities

The present section shows the properties of $\sqsubseteq$ that will be directly used in the proof of undecidability. To this end we introduce ordered derivations. Intuitively, positive rewriting steps will be ordered 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 positions from the root towards the leaves, and then at negative positions, from the leaves upwards.

Definition 4 (ordered derivations).
A sequence of positive steps $\sigma_0 \xRightarrow{p_1}{}^{+}\cdots\xRightarrow{p_n}{}^{+}\sigma_n$ is ordered *if:*

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

$$\begin{aligned}
\sigma_j[\forall\boldsymbol{\delta}\,\alpha\,\boldsymbol{\zeta}\beta\;\varrho]_p \;&\xRightarrow{p}{}^{+}_{i}\; \sigma_j[\forall\boldsymbol{\delta}\,\alpha\,\boldsymbol{\zeta}\;\varrho\{\beta\mapsto\xi_\beta\}]_p\\
&\xRightarrow{p}{}^{+}_{i}\; \sigma_j[\forall\boldsymbol{\delta}\,\boldsymbol{\zeta}\;\varrho\{\beta\mapsto\xi_\beta\}\{\alpha\mapsto\xi_\alpha\}]_p
\end{aligned}$$

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.*

Lemma 8. *If* $\sigma \sqsubseteq \tau$*, then there exists an ordered derivation* $\sigma \Longrightarrow\!\!\!\!\!\Rightarrow^+ \Longrightarrow\!\!\!\!\!\Rightarrow^- \tau$*, of weight equal to the weight of the inequality.*

Proof. Consider a normal derivation $\sigma \Longrightarrow\!\!\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\!\!\Rightarrow^- \tau$ of minimal weight. We observe that every two consecutive steps which are not ordered can be permuted. Since each phase is sorted separately, the middle type ρ does not change and neither does the weight of the derivation. □

Lemma 9. *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 $\sigma \Longrightarrow\!\!\!\!\!\Rightarrow^+ \rho \Longrightarrow\!\!\!\!\!\Rightarrow^- \tau$ be an ordered derivation. Only those steps which have p as a prefix can influence a subterm at position p. Since the derivation is ordered, those steps form two groups, 'in the middle' of both phases. It turns out that those groups of steps can easily be transformed into the desired derivation $\varrho_\sigma \Longrightarrow\!\!\!\!\!\Rightarrow \varrho_\tau$ (or $\varrho_\tau \Longrightarrow\!\!\!\!\!\Rightarrow \varrho_\sigma$ if p is negative). □

Lemma 10. *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.*

Lemma 11. *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 $\sigma\{\alpha \mapsto \xi\} \sqsubseteq \tau\{\alpha \mapsto \xi\}$, then since $\forall\alpha\,\sigma \overset{\epsilon}{\Longrightarrow}{}^+_i \sigma\{\alpha \mapsto \xi\}$, we have

$$\forall\alpha\,\sigma \sqsubseteq \tau\{\alpha \mapsto \xi\} \tag{1}$$

In order to show the opposite implication, suppose inequality (1) holds. Consider its ordered derivation of minimal weight. It begins with some meaningless $\overset{\epsilon}{\Longrightarrow}{}^+$ steps. All they can do is to add some empty quantifiers after the initial $\forall\alpha$. Now the latter must be instantiated:

$$\forall\alpha\,\sigma' \overset{\epsilon}{\Longrightarrow}{}^+_i \sigma'\{\alpha \mapsto \chi\} \tag{2}$$

where σ' stands for $\forall\beta_1 \ldots \beta_n\,\sigma$, and $\beta_1, \ldots, \beta_n$ are the empty quantifiers. By Lemma 9, $\xi \sim \chi$, so also $\sigma\{\alpha \mapsto \xi\} \sim \sigma'\{\alpha \mapsto \chi\}$. By Proposition 7 this yields the desired inequality. Its weight is no greater than the weight of (1) minus the weight of (2), which is strictly positive, when *weight*(ξ) is so. Since *weight*$(\xi) =$ *weight*(χ) we have our claim. □

8 Encoding a machine

A stack register machine $\mathcal{M}$ is a deterministic 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 ℓ from the first main register and executing the instruction assigned to ℓ. If the read label is e the machine halts.

The encoding idea is the following. For each label $\ell \in \Sigma$ we construct a type σ_ℓ embodying the associated instruction. Then we give the label composition rules to encode words contained by the machine's registers. So a given configuration yields $n+2$ types, $\sigma_{V_1}, \sigma_{V_2}, \sigma_{v_1}, \ldots, \sigma_{v_n}$, containing one special free variable γ.

Proposition 12. *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*

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

Proof. The proof is based on two properties of $\sqsubseteq$ which are exactly Lemmas 10 and 11. They correspond to Lemmas 12 and 14 in [9]. See the latter paper for all the details. □

Theorem 2 (Main result).
The relation $\sqsubseteq$ of polymorphic subtyping without distributivity is undecidable.

Proof. This follows from our Proposition 12 and Lemma 16 from the paper of Tiuryn and Urzyczyn [9] that reduces the halting problem of deterministic two-counter automata to the halting problem of stack register machines. □

9 Conclusions

The same proof would also be valid if we removed both **(distr)** and **(quant)** from the definition of subtyping. The only difference would emerge in Section 6, but Proposition 7 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 [5]. Some work has already been done in this field (see [1]) and we believe that further investigation should lead to interesting results.

References

1. M. Benke, "Predicative Polymorphic Subtyping", these proceedings.
2. J. Chrząszcz, "Polymorphic Subtyping Without Distributivity" Technical Report TR98-03(252), Institute of Informatics, Warsaw University, May 1998. URL: `http:\\zls.mimuw.edu.pl\~chrzaszc\papers`

3. J.-Y. Girard, Y. Lafont, P. Taylor. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1989.
4. T. Jim, "System F plus Subsumption Reduces to Mitchell's Subtyping Relation", Manuscript (1995).
5. D. Leivant, "Finitely Stratified Polymorphism", *Information and Computation*, **93** (1), 1991, 93-113.
6. G. Longo, K. Milsted, S. Soloviev, "A Logic of Subtyping", *Proc. 10th IEEE Symp. Logic in Computer Science*, 1995, pp. 292-299.
7. J.C. Mitchell, "Polymorphic Type Inference and Containment", *Information and Computation*, **76** (2-3), 1988, pp. 211-249.
8. J. Tiuryn, "Equational Axiomatization of Bicoercibility for Polymorphic Types", *Proc. Conf. Foundations of Software Technology and Theoretical Computer Science'95*, Bangalore, India; 18-20 Dec 1995. Lecture Notes in Computer Science 1026, pp. 166-179.
9. J. Tiuryn, P. Urzyczyn, "The Subtyping Problem for Second-Order Types is Undecidable" *Proc. 11th IEEE Symp.Logic in Computer Science*, 1996, pp. 74-85.
10. J.B. Wells, "Typability and Type Checking in the Second-Order λ-calculus are Equivalent and Undecidable", *Proc. 9th IEEE Symp. Logic in Computer Science*, 1994, pp. 176-185.
11. J.B. Wells, "The Undecidability of Mitchell's Subtyping Relationship", Technical Report, Computer Science Department, Boston University, Number 95-019, December 10 1995.
12. J.B. Wells, "Typability is Undecidable for F+Eta" Technical Report, Computer Science Department, Boston University, Number 96-022, March 9 1996.