Undecidability of subtyping in System $F_{\leqslant:}$

1 System $F_{\leqslant:}$ and variants

We concern ourselves with a minimal system encapsulating the notion of bounded quantification, it defines the reflexive and transitive subtype relation and uses a maximal type to recover unbounded quantification.

Definition 1.1. *$F_{\leqslant:}$ is the least relation closed under the following.*

$$\frac{}{\Gamma \vdash \tau \leqslant: \tau}\ \textit{Refl} \qquad \frac{}{\Gamma \vdash \tau \leqslant: \top}\ \textit{Top}$$

$$\frac{\Gamma \vdash \tau_1 \leqslant: \tau_2 \qquad \Gamma \vdash \tau_2 \leqslant: \tau_3}{\Gamma \vdash \tau_1 \leqslant: \tau_3}\ \textit{Trans} \qquad \frac{}{\Gamma \vdash \alpha \leqslant: \Gamma(\alpha)}\ \textit{Var}$$

$$\frac{\Gamma \vdash \tau_1 \leqslant: \sigma_1 \qquad \Gamma, \alpha \leqslant: \tau_1 \vdash \sigma_2 \leqslant: \tau_2}{\Gamma \vdash \forall \alpha \leqslant: \sigma_1.\ \sigma_2 \leqslant: \forall \alpha \leqslant: \tau_1.\ \tau_2}\ \textit{All}$$

The contravariance of universal quantifiers makes it possible to *rebound* a variable in the body of the left hand side. This is what makes the subtyping undecidable as it can be proven that a system where the bounds of universal quantifiers are required to be identical has decidable subtyping.

The first refinement made to the system is to make the the relation syntax directed, so at each step there is at most one rule applicable. To achive this the $Refl$ rule is restricted only to variables, while the *Trans* rule is subsumed by the new $NVar$ rule which combines an instance of transitivity.

Definition 1.2. *$F^N_{\leqslant:}$ (normal) is the least relation closed under the following.*

$$\frac{}{\Gamma \vdash \alpha \leqslant: \alpha}\ \textit{NRefl} \qquad \frac{}{\Gamma \vdash \tau \leqslant: \top}\ \textit{NTop}$$

$$\frac{\Gamma \vdash \Gamma(\alpha) \leqslant: \tau}{\Gamma \vdash \alpha \leqslant: \tau}\ \textit{NVar}$$

$$\frac{\Gamma \vdash \tau_1 \leqslant: \sigma_1 \qquad \Gamma, \alpha \leqslant: \tau_1 \vdash \sigma_2 \leqslant: \tau_2}{\Gamma \vdash \forall \alpha \leqslant: \sigma_1.\ \sigma_2 \leqslant: \forall \alpha \leqslant: \tau_1.\ \tau_2}\ \textit{NAll}$$

It is well established that this relation is indeed reflexive and transitive so the proof of the following is straightforward.

Theorem 1.3. *A statement is derivable in $F_{\leqslant:}$ iff it is derivable in $F^N_{\leqslant:}$.*

1.1 Polarized syntax

We now make some syntactical restrictions, terms are classified by whether they are allowed to be used to the left (negative) or right (positive) of the subtyping relation. The following operator is defined to *flip* the sign of a term.

Definition 1.4. $\overline{\tau} := \forall \alpha \leqslant: \tau.\ \alpha$

Lemma 1.5. $\Gamma \vdash \overline{\sigma} \leqslant: \overline{\tau} \iff \Gamma \vdash \tau \leqslant: \sigma$

Additionally we make negative quantified terms unbounded, as they will be rebounded later. The grammar of the positive and negative terms is defined as follows.

$$\tau^+ ::= \top \mid \overline{\tau^-} \mid \forall \alpha \leqslant: \tau^-.\ \tau^+$$

$$\tau^- ::= \alpha \mid \overline{\tau^+} \mid \forall \alpha.\ \tau^-$$

We can check that the rules preserve the signs of terms: if a metavariable appears with a given sign in a premise it has the same sign in the conclusion. We restrict the syntax further to disallow negative terms being compared to quantified terms.

For a given natural number n, the n-positive and n-negative terms are defined by the following grammar.

$$\tau^+ ::= \top \mid \forall \alpha_0 \leqslant: \tau_0^- \dots \alpha_n \leqslant: \tau_n^-.\ \overline{\tau^-}$$

$$\tau^- ::= \alpha \mid \forall \alpha_0 \dots \alpha_n.\ \overline{\tau^+}$$

We can now make a new refinement to the system to make the derivation of judgements deterministic. We drop the signs of terms to avoid notational clutter.

Definition 1.6. *$F^D_{\leqslant:}$ (deterministic) is the least relation closed under the following.*

$$\frac{}{\Gamma \vdash \tau \leqslant: \top}\ \textit{DTop}$$

$$\frac{\Gamma \vdash \Gamma(\alpha) \leqslant: \forall \alpha_0 \leqslant: \sigma_0 \dots \alpha_n \leqslant: \sigma_n.\ \overline{\tau}}{\Gamma \vdash \alpha \leqslant: \forall \alpha_0 \leqslant: \sigma_0 \dots \alpha_n \leqslant: \sigma_n.\ \overline{\tau}}\ \textit{DVar}$$

$$\frac{\Gamma, \alpha_0 \leqslant: \phi_0, \dots, \alpha_n \leqslant: \phi_n \vdash \tau \leqslant: \sigma}{\Gamma \vdash \forall \alpha_0 \dots \alpha_n.\ \overline{\sigma} \leqslant: \forall \alpha_0 \leqslant: \phi_0 \dots \alpha_n \leqslant: \phi_n.\ \overline{\tau}}\ \textit{DAllNeg}$$

Statements of $F^D_{\leqslant:}$ may be regarded as statements of $F^N_{\leqslant:}$ after using the abbreviations for negation and unbounded quantification.

Theorem 1.7. *$F^N_{\leqslant:}$ is a conservative extension of $F^D_{\leqslant:}$; an $F^D_{\leqslant:}$ statement is derivable iff it is derivable when considered as an $F^N_{\leqslant:}$ statement.*

Proof. One direction is immediate, the other follows by induction of $F^N_{\leqslant:}$ derivations whose conclusions are de-abbreviated $F^D_{\leqslant:}$ statements. □

To make the transition to row machines easier we need another variation to keep the context empty so the quantifier rule immediately substitutes the bounds in the body of the statement. This is safe to do because of the restriction on variables to appear only negatively.

1.2 Eager substitution

Definition 1.8. *$F^F_{\leqslant:}$ (flattened) is the least relation closed under the following.*

$$\frac{}{\vdash \tau \leqslant: \top}\ \mathit{FTop}$$

$$\frac{\vdash \tau[\phi_0/\alpha_0 \dots \phi_n/\alpha_n] \leqslant: \sigma[\phi_0/\alpha_0 \dots \phi_n/\alpha_n]}{\vdash \forall \alpha_0 \dots \alpha_n.\, \overline{\sigma} \leqslant: \forall \alpha_0 \leqslant: \phi_0 \dots \alpha_n \leqslant: \phi_n.\, \overline{\tau}}\ \mathit{FAllNeg}$$

In order to show that $F^D_{\leqslant:}$ is a conservative extension of this system we first have to show that $F^D_{\leqslant:}$ indeed supports eager substitution.

Lemma 1.9. *For all closed ϕ_1,..., ϕ_n the following holds in $F^D_{\leqslant:}$*

$$\alpha_1 \leqslant: \phi_1, \dots, \alpha_n \leqslant: \phi_n, \Gamma \vdash \sigma \leqslant: \tau \iff \Gamma[\phi_1/\alpha_1, \dots, \phi_n/\alpha_n] \vdash \sigma[\phi_1/\alpha_1, \dots, \phi_n/\alpha_n] \leqslant: \tau[\phi_1/\alpha_1, \dots, \phi_n/\alpha_n]$$

Proof. Both sides follow by induction on the derivation. □

Theorem 1.10. *$\vdash \sigma \leqslant: \tau$ is derivable in $F^D_{\leqslant:}$ iff it is derivable in $F^F_{\leqslant:}$*

Proof. By induction on the derivation using the previous lemma for the quantifier case. □

2 Rowing Machines

We now turn our attention to an intermediate model that encapsulates the eager substitution of the last system, the rowing machines.

Definition 2.1. *For a given natural number n a rowing machine is a closed tuple of registers*

$$\langle \rho_1 \dots \rho_n \rangle$$

each of which is a row of width n defined by the following grammar

$$\rho ::= \alpha \mid [\alpha_1 \dots \alpha_n]\langle \rho_1 \dots \rho_n \rangle \mid \texttt{Halt}$$

In the term $[\alpha_1 \dots \alpha_n]\langle \rho_1 \dots \rho_n \rangle$ the variables α_i are binding occurrences whose scope is all of the rows ρ_j. A row is closed if it contains no free variables, and a rowing machine is closed if it consists of closed rows.

Definition 2.2. *The step function is the partial function defined by the following mapping.*
If $\rho_1 = [\alpha_1 \dots \alpha_n]\langle \rho'_1 \dots \rho'_n \rangle$

$$\langle \rho_1 \dots \rho_n \rangle \mapsto \langle \rho'_1[\rho_1/\alpha_1 \dots \rho_n/\alpha_n] \dots \rho'_1[\rho_1/\alpha_1 \dots \rho_n/\alpha_n] \rangle$$

We write $M \to_R M'$ to denote that the rowing machine M is mapped to M' by the step function.

Because rowing machines consists of closed rows the first element cannot be a variable. Additionally the step function is undefined when the first element is `Halt`, as expected.

Definition 2.3. *A rowing machine M halts if for some ρ_2, ..., ρ_n*

$$M \to_R^* \langle \texttt{Halt}, \rho_2 \dots \rho_n \rangle$$

We can now encode rowing machines as subtyping problems in an appropriate way; if the rowing machine has `Halt` as its first row then it it should be encoded as a statement that reaches a subproblem with $\top$ on the right, alternatively if the rowing machine steps into another then its encoded should be a judgment that reaches the encoding of the the second machine.

Abusing notation we write $\mathcal{F}$ for both the encoding of rowing machines and rows.

Definition 2.4. *The encoding of rows as $F^F_{\leqslant:}$ terms is defined as follows.*

$$\begin{aligned}
\mathcal{F}(\alpha_i) &:= \alpha_i \\
\mathcal{F}(\texttt{Halt}) &:= \forall \gamma_0, \alpha_1 \dots \alpha_n.\ \overline{\top} \\
\mathcal{F}([\alpha_1 \dots \alpha_n]\langle \rho_1 \dots \rho_n \rangle) &:= \forall \gamma_0, \alpha_1 \dots \alpha_n.\ \overline{\forall \gamma_0' \leqslant: \gamma_0, \alpha_1' \leqslant: \mathcal{F}(\rho_1) \dots \alpha_n' \leqslant: \mathcal{F}(\rho_n).\ \overline{\mathcal{F}(\rho_1)}}
\end{aligned}$$

Definition 2.5. *The encoding of a row machine as an $F^F_{\leqslant:}$ statement is defined as follows.*

$$\mathcal{F}(\langle \rho_1 \dots \rho_n \rangle) := \vdash \sigma \leqslant: \forall \gamma_0 \leqslant: \sigma, \gamma_1 \leqslant: \mathcal{F}(\rho_1) \dots \gamma_n \leqslant: \mathcal{F}(\rho_n).\ \overline{\mathcal{F}(\rho_1)}$$

where $\sigma := \forall \gamma_0 \dots \gamma_n.\ \overline{\forall \gamma_0' \leqslant: \gamma_0 \dots \gamma_n' \leqslant: \gamma_n.\ \overline{\gamma_0}}$

It is easy to check that in fact $\mathcal{F}(R)$ is a closed $F^F_{\leqslant:}$ statement for any rowing machine R. We can prove the translation has the expected property.

Lemma 2.6. *If $R \to_R R'$ then $\mathcal{F}(R)$ is derivable from $\mathcal{F}(R')$*

Proof. Let $R = \langle \rho_1 \dots \rho_n \rangle$, because R steps to R' we now that ρ_1 is of the form $[\alpha_1 \dots \alpha_n]\langle \rho_1' \dots \rho_n' \rangle$, and $R' = \langle \rho_1'[\rho_1/\alpha_1 \dots \rho_n/\alpha_n] \dots \rho_1'[\rho_1/\alpha_1 \dots \rho_n/\alpha_n] \rangle$

$$\dfrac{\dfrac{\mathcal{F}(R') \equiv \vdash \sigma \leqslant: \forall \gamma_0 \leqslant: \sigma, \gamma_1 \leqslant: \mathcal{F}(\rho_1')[\theta] \dots \gamma_n \leqslant: \mathcal{F}(\rho_n')[\theta].\ \overline{\mathcal{F}(\rho_1')[\theta]}}{\vdash \mathcal{F}(\rho_1)[\sigma/\gamma_0, \mathcal{F}(\rho_1)/\gamma_1 \dots \mathcal{F}(\rho_n)/\gamma_n] \leqslant: \forall \gamma_0' \leqslant: \sigma, \gamma_1' \leqslant: \mathcal{F}(\rho_1) \dots \gamma_n' \leqslant: \mathcal{F}(\rho_n).\overline{\sigma}}\ \text{FAllNeg}}{\mathcal{F}(R) \equiv \vdash \sigma \leqslant: \forall \gamma_0 \leqslant: \sigma, \gamma_1 \leqslant: \mathcal{F}(\rho_1) \dots \gamma_n \leqslant: \mathcal{F}(\rho_n).\ \overline{\mathcal{F}(\rho_1)}}\ \text{FAllNeg}$$

where θ is the substitution $[\mathcal{F}(\rho_1)/\alpha_1 \dots \mathcal{F}(\rho_n)/\alpha_n]$. □

Lemma 2.7. *$\mathcal{F}(\langle \texttt{Halt}, \rho_2 \dots \rho_n \rangle)$ is derivable for all ρ_2,..., ρ_n*

Proof.

$$\dfrac{\dfrac{\dfrac{}{\vdash \sigma \leqslant: \top}\ \text{FTop}}{\vdash (\forall \gamma_0, \alpha_1 \dots \alpha_n.\ \overline{\top}) \leqslant: (\forall \gamma_0' \leqslant: \sigma \dots \gamma_n' \leqslant: \mathcal{F}(\rho_n).\ \overline{\sigma})}\ \text{FAllNeg}}{\mathcal{F}(\langle \texttt{Halt}, \rho_2 \dots \rho_n \rangle) \equiv \vdash \sigma \leqslant: \forall \gamma_0 \leqslant: \sigma, \gamma_1 \leqslant: \mathcal{F}(\texttt{Halt}) \dots \gamma_n \leqslant: \mathcal{F}(\rho_n).\ \overline{\mathcal{F}(\texttt{Halt})}}\ \text{FAllNeg}$$

□

Theorem 2.8. *The rowing machine R halts iff $\mathcal{F}(R)$ is derivable.*

We use some abbreviations for denoting rows.

- When the symbol $-$ appears as the ith component of a row $[\alpha_1 \dots \alpha_n]\langle \rho_1 \dots \rho_n \rangle$ it stands for the variable α_i.
- Instead of using variable names we use their indices, $\#i$ denotes the ith bound variable of the row in which it appears, $\#\#i$ denotes the ith bound variable of the row enclosing the one in which it appears, and so on.

For example, we can abbreviate the following nested row as follows.

$$[\alpha_1 \dots \alpha_3]\langle \alpha_1, [\beta_1 \dots \beta_3]\langle \alpha_1, \beta_1, \beta_3 \rangle, \alpha_1 \rangle \equiv \langle -, \langle \#\#1, \#1, - \rangle, \#1 \rangle$$

3 Two Counter Machines

A 2-counter machine consists of a program counter and two registers (PC, A, B), and a program which is a list of instructions of the form:

$$\begin{aligned} \mathit{instr} ::= & \mathtt{inc}_A n \mid \mathtt{inc}_B n \\ & \mid \mathtt{dec}_A n/m \mid \mathtt{dec}_B n/m \\ & \mid \mathtt{Halt} \end{aligned}$$

Definition 3.1. *For a program $I_1, \dots, I_n$ the step function is the partial function defined by the following*

$$\begin{aligned} (\mathtt{inc}_A n, A, B) &\mapsto (I_n, SA, B) \\ (\mathtt{inc}_B n, A, B) &\mapsto (I_n, A, SB) \\ (\mathtt{dec}_A n/m, 0, B) &\mapsto (I_n, 0, B) \\ (\mathtt{dec}_A n/m, SA, B) &\mapsto (I_m, A, B) \\ (\mathtt{dec}_B n/m, A, 0) &\mapsto (I_n, A, 0) \\ (\mathtt{dec}_B n/m, A, SB) &\mapsto (I_m, A, B) \end{aligned}$$

We write $M \to_{2CM} M'$ to denote that the 2-counter machine M is mapped to M' by the step function.

Definition 3.2. *A 2-counter machine M halts if for some A and B*

$$M \to^*_{2CM} (\mathtt{Halt}, A, B)$$

In order to encode 2-counter machines as rowing machines we need an encoding of instructions and of natural numbers in each register. For a program of length n the encodings are rows of width $n + 5$, the first 5 registers store PC, A, B, and the branching addresses for the decreasing instruction, the last n registers store the encoding of the program. Abusing notation we use $\mathcal{R}$ for all the encodings.

Definition 3.3. *The encoding of instructions is defined as:*

$$\begin{aligned}
\mathcal{R}(\texttt{inc}_A m) &:= \langle \#m+5, \langle \#5, \#\#2, -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle, -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle \\
\mathcal{R}(\texttt{inc}_B m) &:= \langle \#m+5, -, \langle \#5, -, \#\#3, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle \\
\mathcal{R}(\texttt{dec}_A m/n) &:= \langle \#2, -, -, \#m+5, \#n+5, -\cdots- \rangle \\
\mathcal{R}(\texttt{dec}_B m/n) &:= \langle \#3, -, -, \#m+5, \#n+5, -\cdots- \rangle \\
\mathcal{R}(\texttt{Halt}) &:= \langle \texttt{Halt}, -, -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle
\end{aligned}$$

The encoding of the contents of each register is defined as:

$$\begin{aligned}
\mathcal{R}_A(0) &:= \langle \#4, -, -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle \\
\mathcal{R}_A(n+1) &:= \langle \#5, \mathcal{R}_A(n), -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle
\end{aligned}$$

$$\begin{aligned}
\mathcal{R}_B(0) &:= \langle \#4, -, -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle \\
\mathcal{R}_B(n+1) &:= \langle \#5, -, \mathcal{R}_B(n), \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle
\end{aligned}$$

The encoding of a 2-counter machine and a program is defined as:

$$\mathcal{R}(PC, A, B, I_1, \ldots, I_n) := \langle \mathcal{R}(PC), \mathcal{R}_A(A), \mathcal{R}(B), \mathcal{R}_B(I_1), \ldots, \mathcal{R}(I_n) \rangle$$

Lemma 3.4. *If $M \to_{2CM} M'$ then $\mathcal{R}(M) \to_R^+ \mathcal{R}(M')$*

Proof. By case analysis on PC, as the machine steps it cannot be $\texttt{Halt}$.
Case $PC = \texttt{inc}_A n$, so $M' = (I_n, A+1, B)$

$$\begin{aligned}
\mathcal{R}(M) \equiv& \langle \langle \#m+5, \langle \#5, \#\#2, -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle, -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle, \\
& \mathcal{R}_A(A), \mathcal{R}_B(B), \texttt{Halt}, \texttt{Halt}, \mathcal{R}(I_1)\ldots\mathcal{R}(I_w) \rangle \\
\to_R& \langle \mathcal{R}(I_m), \langle \#5, \mathcal{R}_A(A), -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle, \mathcal{R}_B(B), \texttt{Halt}, \texttt{Halt}, \mathcal{R}(I_1)\ldots\mathcal{R}(I_w) \rangle \\
\equiv& \langle \mathcal{R}(I_m), \mathcal{R}_A(A+1), \mathcal{R}_B(B), \texttt{Halt}, \texttt{Halt}, \mathcal{R}(I_1)\ldots\mathcal{R}(I_w) \rangle \equiv \mathcal{R}(M')
\end{aligned}$$

Case $PC = \texttt{dec}_A m/n$ and $A = 0$, so $M' = (I_m, 0, B)$

$$\begin{aligned}
\mathcal{R}(M) \equiv& \langle \langle \#2, -, -, \#m+5, \#n+5, -\cdots- \rangle, \mathcal{R}_A(0), \mathcal{R}_B(B), \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle \\
\to_R& \langle \langle \#4, -, -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle, \mathcal{R}_A(0), \mathcal{R}_B(B), \mathcal{R}(I_m), \mathcal{R}(I_n), \mathcal{R}(I_1)\ldots\mathcal{R}(I_w) \rangle \\
\to_R& \langle \mathcal{R}(I_m), \mathcal{R}_A(0), \mathcal{R}_B(B), \texttt{Halt}, \texttt{Halt}, \mathcal{R}(I_1)\ldots\mathcal{R}(I_w) \rangle \equiv \mathcal{R}(M')
\end{aligned}$$

Case $PC = \texttt{dec}_A m/n$ and $A = A' + 1$, so $M' = (I_n, A', B)$

$$\begin{aligned}
\mathcal{R}(M) \equiv& \langle \langle \#2, -, -, \#m+5, \#n+5, -\cdots- \rangle, \mathcal{R}_A(A'+1), \mathcal{R}_B(B), \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle \\
\to_R& \langle \langle \#5, \mathcal{R}_A(A'), -, \texttt{Halt}, \texttt{Halt}, -\cdots- \rangle, \mathcal{R}_A(A'+1), \mathcal{R}_B(B), \mathcal{R}(I_m), \mathcal{R}(I_n), \mathcal{R}(I_1)\ldots\mathcal{R}(I_w) \rangle \\
\to_R& \langle \mathcal{R}(I_n), \mathcal{R}_A(A'), \mathcal{R}_B(B), \texttt{Halt}, \texttt{Halt}, \mathcal{R}(I_1)\ldots\mathcal{R}(I_w) \rangle \equiv \mathcal{R}(M')
\end{aligned}$$

The cases for the register B are similar. □

Lemma 3.5. *For all A, B, and programs $I_1, \ldots, I_n$*

$$\mathcal{R}(\texttt{Halt}, A, B, I_1, \ldots, I_n) \to_R \langle \texttt{Halt}, \mathcal{R}_A(A), \mathcal{R}_B(B), \texttt{Halt}, \texttt{Halt}, \mathcal{R}(I_1)\ldots\mathcal{R}(I_w) \rangle$$

Theorem 3.6. *M halts iff $\mathcal{R}(M)$ halts.*

4 Undecidability

Theorem 4.1. *Subtyping in $F_{\leqslant:}$ is undecidable.*

Proof. Assuming we have a decider, we get a decider for any 2-counter machine T:

T halts
iff $\mathcal{R}(T)$ halts (Theorem 3.6)
iff $\mathcal{F}(\mathcal{R}(T))$ is derivable in $F^{F}_{\leqslant:}$ (Theorem 2.8)
iff $\mathcal{F}(\mathcal{R}(T))$ is derivable in $F^{D}_{\leqslant:}$ (Theorem 1.10)
iff $\mathcal{F}(\mathcal{R}(T))$ is derivable in $F^{N}_{\leqslant:}$ (Theorem 1.7)
iff $\mathcal{F}(\mathcal{R}(T))$ is derivable in $F_{\leqslant:}$ (Theorem 1.3)
□

References

[Pierce, 1994] Pierce, B. C. (1994). Bounded quantification is undecidable. *Information and Computation*, pages 131–165.

[Pierce, 2002] Pierce, B. C. (2002). *Types and programming languages*. MIT Press.