Mechanized undecidability of subtyping in System F

Roberto Álvarez
Advisor: Yannick Forster

Saarland University
Programming Systems Lab

Final talk of Master's seminar

30.09.2021

Recap: System $F_{\leqslant:}$

Combines type polymorphism with subtyping.

Terms and types:

$$s, t ::= x \mid \lambda_{x:\tau}.\, t \mid \Lambda_{\alpha\leqslant:\tau}.\, t \mid t\, s \mid t\, \tau$$

$$\sigma, \tau ::= \alpha \mid \sigma \to \tau \mid \forall_{\alpha\leqslant:\sigma}.\, \tau \mid \top$$

Recap: System $F_{\leqslant:}$

Combines type polymorphism with subtyping.

Terms and types:

$$s, t ::= x \mid \lambda_{x:\tau}.\, t \mid \Lambda_{\alpha \leqslant: \tau}.\, t \mid t\, s \mid t\, \tau$$

$$\sigma, \tau ::= \alpha \mid \sigma \to \tau \mid \forall_{\alpha \leqslant: \sigma}.\, \tau \mid \top$$

Recap: System $F_{\leqslant:}$

Combines type polymorphism with subtyping.

Terms and types:

$$s, t ::= x \mid \lambda_{x:\tau}.\, t \mid \Lambda_{\alpha \leqslant: \tau}.\, t \mid t\, s \mid t\, \tau$$

$$\sigma, \tau ::= \alpha \mid \sigma \to \tau \mid \forall_{\alpha \leqslant: \sigma}.\, \tau \mid \top$$

Unbounded quantification can be defined with $\top$:

$$\forall \alpha.\, \tau := \forall_{\alpha \leqslant: \top}.\, \tau$$

Recap: System $F_{\leqslant:}$

1985 System $F_{\leqslant:}$ is first introduced by Cardelli and Wegner. They show coherence of the typechecking algorithm.

[1] Ghelli's post

Recap: System $F_{\leqslant:}$

1985 System $F_{\leqslant:}$ is first introduced by Cardelli and Wegner. They show coherence of the typechecking algorithm.

1990 Ghelli gives a proof of termination.
The proof turns out to be "full of typos"[1].

[1] Ghelli's post

Recap: System $F_{\leqslant:}$

1985 System $F_{\leqslant:}$ is first introduced by Cardelli and Wegner. They show coherence of the typechecking algorithm.

1990 Ghelli gives a proof of termination.
The proof turns out to be "full of typos"[1].

1992 Ghelli gives a counterexample.

[1] Ghelli's post

Recap: System $F_{\leqslant:}$

1985 System $F_{\leqslant:}$ is first introduced by Cardelli and Wegner. They show coherence of the typechecking algorithm.

1990 Ghelli gives a proof of termination.
The proof turns out to be "full of typos"[1].

1992 Ghelli gives a counterexample.

1994 Pierce gives a proof of undecidability.

[1] Ghelli's post

Recap: System $F_{\leqslant:}$

1985 System $F_{\leqslant:}$ is first introduced by Cardelli and Wegner. They show coherence of the typechecking algorithm.

1990 Ghelli gives a proof of termination.
The proof turns out to be ”full of typos”[1].

1992 Ghelli gives a counterexample.

1994 Pierce gives a proof of undecidability.

2021 Pierces's proof is mechanized.

[1] Ghelli's post

Recap: System $F_{\leqslant:}$

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

we say that σ_2 gets *rebounded*.

Recap: System $F_{\leqslant:}$

$$\frac{\Gamma \vdash \tau_1 \leqslant: \sigma_1 \qquad \Gamma, \alpha \leqslant: \tau_1 \vdash \sigma_2 \leqslant: \tau_2}{\Gamma \vdash \forall_{\alpha \leqslant: \sigma_1}.\, \sigma_2 \;\leqslant:\; \forall_{\alpha \leqslant: \tau_1}.\, \tau_2}\ \text{All}$$

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

$$\frac{\Gamma \vdash \sigma \leqslant: \phi \qquad \Gamma \vdash \phi \leqslant: \tau}{\Gamma \vdash \sigma \leqslant: \tau}\ \text{Trans} \qquad \frac{}{\Gamma \vdash \alpha \leqslant: \Gamma(\alpha)}\ \text{Var}$$

Recap: System $F_{\leqslant:}$

$$\frac{\Gamma \vdash \tau_1 \leqslant: \sigma_1 \qquad \Gamma, \alpha \leqslant: \tau_1 \vdash \sigma_2 \leqslant: \tau_2}{\Gamma \vdash \forall_{\alpha \leqslant: \sigma_1}.\, \sigma_2 \;\leqslant:\; \forall_{\alpha \leqslant: \tau_1}.\, \tau_2}\ \textsf{All}$$

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

$$\frac{\Gamma \vdash \sigma \leqslant: \phi \qquad \Gamma \vdash \phi \leqslant: \tau}{\Gamma \vdash \sigma \leqslant: \tau}\ \textsf{Trans} \qquad \frac{}{\Gamma \vdash \alpha \leqslant: \Gamma(\alpha)}\ \textsf{Var}$$

$F_{\leqslant:}$ subtyping:
Given arbitrary Γ, σ and τ, is there a derivation of $\Gamma \vdash \sigma \leqslant: \tau$?

Recap: undecidability

Theorem

$F_{\leqslant:}$ subtyping is synthetically undecidable.

Recap: undecidability

Theorem

$F_{\leqslant:}$ subtyping is synthetically undecidable.

Proof.

By a chain of many-one reductions, Pierce [1994]:
2CM halting $\preceq_m$ RM halting $\preceq_m \cdots \preceq_m F_{\leqslant:}$ subtyping

□

Recap: undecidability

Theorem

$F_{\leqslant:}$ subtyping is synthetically undecidable.

Proof.

By a chain of many-one reductions, Pierce [1994]:

$$\underbrace{\text{2CM halting} \preceq_m \text{RM halting}}_{\text{mechanized before first talk}} \quad \underbrace{\preceq_m \cdots \preceq_m F_{\leqslant:} \text{ subtyping}}_{\text{partially mechanized since}}$$

□

Recap: undecidability

Theorem
$F_{\leqslant:}$ subtyping is synthetically undecidable.

Proof.
By a chain of many-one reductions, Pierce [1994]:

$$\underbrace{\text{2CM halting} \preceq_m \text{RM halting}}_{\text{mechanized before first talk}} \quad \underbrace{\preceq_m \cdots \preceq_m F_{\leqslant:} \text{ subtyping}}_{\text{partially mechanized since}}$$

□

To show RM halting $\preceq_m F_{\leqslant:}$ subtyping Pierce shows:

$$R \text{ halts} \iff \vdash \sigma \leqslant : \mathcal{T}(R)$$

for a concrete σ independent of R.

Overview

$$R \text{ halts} \iff \vdash \sigma \leqslant : \mathcal{T}(R)$$

($\Rightarrow$) By induction on the trace, in order to encode the stepping of the machine we need:

Overview

$$R \text{ halts} \iff \vdash \sigma \leqslant : \mathcal{T}(R)$$

($\Rightarrow$) By induction on the trace, in order to encode the stepping of the machine we need:

- To rebound the right hand side with an operator that *flips* inequalities using contravariance:

$$\overline{\tau} := \forall_{\alpha \leqslant : \tau}.\alpha$$

$$\Gamma \vdash \overline{\sigma} \leqslant : \overline{\tau} \iff \Gamma \vdash \tau \leqslant : \sigma \tag{1}$$

Overview

$$R \text{ halts} \iff \vdash \sigma \leqslant : \mathcal{T}(R)$$

($\Rightarrow$) By induction on the trace, in order to encode the stepping of the machine we need:

- To rebound the right hand side with an operator that *flips* inequalities using contravariance:

$$\overline{\tau} := \forall_{\alpha \leqslant : \tau}.\alpha$$

$$\Gamma \vdash \overline{\sigma} \leqslant : \overline{\tau} \iff \Gamma \vdash \tau \leqslant : \sigma \tag{1}$$

- To substitute variables eagerly, as the machine does:

$$\alpha \leqslant : \phi \vdash \sigma \leqslant : \tau \iff \vdash \sigma[\phi/\alpha] \leqslant : \tau[\phi/\alpha] \tag{2}$$

  Does not hold in general, e.g. with $\phi = \sigma = \top$ and $\tau = \alpha$.

Overview

$$R \text{ halts} \iff \vdash \sigma \leqslant : \mathcal{T}(R)$$

($\Rightarrow$) By induction on the trace, in order to encode the stepping of the machine we need:

- Flip property:

$$\Gamma \vdash \overline{\sigma} \leqslant : \overline{\tau} \iff \Gamma \vdash \tau \leqslant : \sigma \tag{1}$$

- Eager substitution:

$$\alpha \leqslant : \phi \vdash \sigma \leqslant : \tau \iff \vdash \sigma[\phi/\alpha] \leqslant : \tau[\phi/\alpha] \tag{2}$$

Overview

$$R \text{ halts} \iff \vdash \sigma \leqslant : \mathcal{T}(R)$$

($\Leftarrow$) We need to analyze the derivation, however:

- Transitivity is too general, there might be infinitely many derivations.

Overview

$$R \text{ halts} \iff \vdash \sigma \leqslant : \mathcal{T}(R)$$

($\Leftarrow$) We need to analyze the derivation, however:

- Transitivity is too general, there might be infinitely many derivations.
- We need to obtain derivations deterministically, to match the behaviour of the machine.

Overview

$$R \text{ halts} \iff \vdash \sigma \leqslant : \mathcal{T}(R)$$

($\Leftarrow$) We need to analyze the derivation, however:

- Transitivity is too general, there might be infinitely many derivations.
- We need to obtain derivations deterministically, to match the behaviour of the machine.
- The types are too general; we need an invariant on the syntax. We only care about types of the form of translated machines.

Overview

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

Pierce defines the intermediate systems to address the requirements:

Overview

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

Pierce defines the intermediate systems to address the requirements:

$F^N_{\leqslant:}$ Restricted transitivity and flip property.

Overview

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

Pierce defines the intermediate systems to address the requirements:

- $F^N_{\leqslant:}$ Restricted transitivity and flip property.
- $F^D_{\leqslant:}$ Deterministic subtyping and syntactic invariants.

Overview

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

Pierce defines the intermediate systems to address the requirements:

- $F^N_{\leqslant:}$ Restricted transitivity and flip property.
- $F^D_{\leqslant:}$ Deterministic subtyping and syntactic invariants.
- $F^F_{\leqslant:}$ Eager substitution.

Overview

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

Pierce defines the intermediate systems to address the requirements:

- $F^N_{\leqslant:}$ Restricted transitivity and flip property.
- $F^D_{\leqslant:}$ Deterministic subtyping and syntactic invariants.
- $F^F_{\leqslant:}$ Eager substitution.

The systems are implemented with deBrujin indices, however are presented with named variables.

System $F^N_{\leqslant:}$ (normal)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m \boxed{F^N_{\leqslant:} \preceq_m F_{\leqslant:}}$$

Makes subtyping syntax directed:

$$\frac{}{\Gamma \vdash_N \alpha \leqslant: \alpha}\ \mathsf{NRefl} \qquad \frac{\Gamma \vdash_N \Gamma\ \alpha \leqslant: \tau}{\Gamma \vdash_N \alpha \leqslant: \tau}\ \mathsf{NVar}$$

System $F^N_{\leqslant:}$ (normal)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m \boxed{F^N_{\leqslant:} \preceq_m F_{\leqslant:}}$$

Makes subtyping syntax directed:

$$\frac{}{\Gamma \vdash_N \alpha \leqslant: \alpha}\ \mathsf{NRefl} \qquad \frac{\Gamma \vdash_N \Gamma\ \alpha \leqslant: \tau}{\Gamma \vdash_N \alpha \leqslant: \tau}\ \mathsf{NVar}$$

Theorem 1

$\Gamma \vdash_N \sigma \leqslant: \tau \iff \Gamma \vdash \sigma \leqslant: \tau$

System $F^N_{\leqslant:}$ (normal)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m \boxed{F^N_{\leqslant:} \preceq_m F_{\leqslant:}}$$

Makes subtyping syntax directed:

$$\frac{}{\Gamma \vdash_N \alpha \leqslant: \alpha}\ \mathsf{NRefl} \qquad \frac{\Gamma \vdash_N \Gamma\ \alpha \leqslant: \tau}{\Gamma \vdash_N \alpha \leqslant: \tau}\ \mathsf{NVar}$$

Theorem 1

$\Gamma \vdash_N \sigma \leqslant: \tau \iff \Gamma \vdash \sigma \leqslant: \tau$

The flip property is now immediate.

Lemma 2

$\Gamma \vdash_N \overline{\sigma} \leqslant: \overline{\tau} \iff \Gamma \vdash_N \tau \leqslant: \sigma$

System $F^N_{\leqslant:}$ (normal)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m \boxed{F^N_{\leqslant:} \preceq_m F_{\leqslant:}}$$

Makes subtyping syntax directed:

$$\frac{}{\Gamma \vdash^0_N \alpha \leqslant: \alpha}\ \mathsf{NRefl} \qquad \frac{\Gamma \vdash^i_N \Gamma\ \alpha \leqslant: \tau}{\Gamma \vdash^{\mathsf{S}i}_N \alpha \leqslant: \tau}\ \mathsf{NVar}$$

Theorem 1

$(\exists i.\ \Gamma \vdash^i_N \sigma \leqslant: \tau) \iff \Gamma \vdash \sigma \leqslant: \tau$

The flip property is now immediate.

Lemma 2

$\Gamma \vdash^{\mathsf{S}i}_N \overline{\sigma} \leqslant: \overline{\tau} \iff \Gamma \vdash^i_N \tau \leqslant: \sigma$

Later we'll need the height of the derivations.

System $F^{D}_{\leqslant:}$ (deterministic)

$$\mathsf{RM} \preceq_m F^{F}_{\leqslant:} \preceq_m \boxed{F^{D}_{\leqslant:} \preceq_m F^{N}_{\leqslant:}} \preceq_m F_{\leqslant:}$$

The *polarized* syntax classifies positive and negative terms:

$$\tau^{+} ::= \top \mid \forall_{\alpha_0 \leqslant: \tau_0^{-}, \ldots, \alpha_w \leqslant: \tau_w^{-}}.\, \overline{\tau^{-}}$$

$$\tau^{-} ::= \alpha \mid \forall_{\alpha_0, \ldots, \alpha_w}.\, \overline{\tau^{+}}$$

System $F^{D}_{\leqslant:}$ (deterministic)

$$\mathsf{RM} \preceq_m F^{F}_{\leqslant:} \preceq_m \boxed{F^{D}_{\leqslant:} \preceq_m F^{N}_{\leqslant:}} \preceq_m F_{\leqslant:}$$

The *polarized* syntax classifies positive and negative terms:

$$\tau^{+} ::= \top \mid \forall_{\alpha_0 \leqslant: \tau_0^{-}, \ldots, \alpha_w \leqslant: \tau_w^{-}}.\, \overline{\tau^{-}}$$
$$\tau^{-} ::= \alpha \mid \forall_{\alpha_0, \ldots, \alpha_w}.\, \overline{\tau^{+}}$$

The *polyadic* binders are the syntactic invariant required: machines have a constant number of registers that are updated simultaneously.

System $F^D_{\leqslant:}$ (deterministic)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m \boxed{F^D_{\leqslant:} \preceq_m F^N_{\leqslant:}} \preceq_m F_{\leqslant:}$$

The *polarized* syntax classifies positive and negative terms:

$$\tau^+ ::= \top \mid \forall_{\alpha_0 \leqslant: \tau_0^-, \ldots, \alpha_w \leqslant: \tau_w^-}. \overline{\tau^-}$$
$$\tau^- ::= \alpha \mid \forall_{\alpha_0, \ldots, \alpha_w}. \overline{\tau^+}$$

The *polyadic* binders are the syntactic invariant required: machines have a constant number of registers that are updated simultaneously.

New quantifier rule:

$$\frac{\Gamma, \alpha_0 \leqslant: \phi_0^-, \ldots \alpha_w \leqslant: \phi_w^- \vdash^i_D \tau^- \leqslant: \sigma^+}{\Gamma \vdash^{\mathsf{S}i}_D \forall_{\alpha_0, \ldots, \alpha_w}. \overline{\sigma^+} \leqslant: \forall_{\alpha_0 \leqslant: \phi_0^-, \ldots, \alpha_w \leqslant: \phi_w^-}. \overline{\tau^-}} \ \mathsf{DAllFlip}$$

System $F^D_{\leqslant:}$ (deterministic)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m \boxed{F^D_{\leqslant:} \preceq_m F^N_{\leqslant:}} \preceq_m F_{\leqslant:}$$

We need a translation $[\![-]\!]$ from *well-scoped polyadic* syntax to *unscoped* syntax:

$$[\![\mathsf{var}_D\ i\ j]\!] = \mathsf{var}_N\ (\widehat{i} + w * \widehat{j})$$

where $i : \mathbb{I}^w$ and $j : \mathbb{I}^n$ for some n.

System $F^D_{\leqslant:}$ (deterministic)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m \boxed{F^D_{\leqslant:} \preceq_m F^N_{\leqslant:}} \preceq_m F_{\leqslant:}$$

We need a translation $[\![-]\!]$ from *well-scoped polyadic* syntax to *unscoped* syntax:

$$[\![\mathsf{var}_D\ i\ j]\!] = \mathsf{var}_N\ (\widehat{i} + w * \widehat{j})$$

where $i : \mathbb{I}^w$ and $j : \mathbb{I}^n$ for some n.

Translating renamings gets complicated, there are lemmas yet to be completed!

System $F^D_{\leqslant:}$ (deterministic)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m \boxed{F^D_{\leqslant:} \preceq_m F^N_{\leqslant:}} \preceq_m F_{\leqslant:}$$

Theorem

$(\exists i.\, \Gamma \vdash^i_D \sigma \leqslant: \tau) \iff (\exists j.\, [\![\Gamma]\!] \vdash^j_N [\![\sigma]\!] \leqslant: [\![\tau]\!])$

Proof.

($\Rightarrow$) By induction on the derivation.

System $F^D_{\leqslant:}$ (deterministic)

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m \boxed{F^D_{\leqslant:} \preceq_m F^N_{\leqslant:}} \preceq_m F_{\leqslant:}$$

Theorem

$(\exists i.\, \Gamma \vdash^i_D \sigma \leqslant: \tau) \iff (\exists j.\, [\![\Gamma]\!] \vdash^j_N [\![\sigma]\!] \leqslant: [\![\tau]\!])$

Proof.

($\Rightarrow$) By induction on the derivation.

($\Leftarrow$) The new quantifier rule corresponds to $w+1$ uses of the old rule, therefore we use complete induction on the height of the derivation.

$$\dfrac{\dfrac{\vdots}{\vdash^i_D}}{\vdash^{\mathsf{S}i}_D}\,\mathsf{DAllFlip}\,\dfrac{\dfrac{\vdots}{\vdash^{j-w-1}_N}}{\dfrac{\vdots}{\vdash^j_N}\,\mathsf{NAll}}\,\mathsf{NAll}$$

System $F^D_{\leqslant:}$ (deterministic)

One can already show a generalization of eager substitution:

Lemma 3

For all i there is a j such that

$$\alpha_0 \leqslant: \phi_0, \ldots, \alpha_w \leqslant: \phi_w, \Gamma \vdash^i_D \sigma \leqslant: \tau$$

$$\Longleftrightarrow$$

$$\Gamma[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w] \vdash^j_D \sigma[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w] \leqslant: \tau[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w]$$

and $j \leq i$.

System $F^D_{\leqslant:}$ (deterministic)

One can already show a generalization of eager substitution:

Lemma 3

For all i there is a j such that

$$\alpha_0 \leqslant: \phi_0, \ldots, \alpha_w \leqslant: \phi_w, \Gamma \vdash^i_D \sigma \leqslant: \tau$$

$$\Longleftrightarrow$$

$$\Gamma[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w] \vdash^j_D \sigma[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w] \leqslant: \tau[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w]$$

and $j \leq i$.

Proof.

Both directions follow by induction.
The proof involves substituting the closed types that were first introduced in a context, this motivates the use of well-scoped syntax. □

System $F^F_{\leqslant:}$ (flattened)

$$\mathsf{RM} \preceq_m \boxed{F^F_{\leqslant:} \preceq_m F^D_{\leqslant:}} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

The final variant incorporates eager substitution in the quantifier rule:

$$\frac{\vdash^i_F \tau[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w] \leqslant: \sigma[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w]}{\vdash^{\mathsf{S}i}_F \forall_{\alpha_0 \leqslant: \top, \ldots, \alpha_w \leqslant: \top}.\, \overline{\sigma} \leqslant: \forall_{\alpha_0 \leqslant: \phi_0, \ldots, \alpha_w \leqslant: \phi_w}.\, \overline{\tau}}\ \mathsf{FAllFlip}$$

Theorem 4

$(\exists i.\ \vdash^i_F \sigma \leqslant: \tau) \iff (\exists j.\ \vdash^j_D \sigma \leqslant: \tau)$

System $F^F_{\leqslant:}$ (flattened)

$$\mathsf{RM} \preceq_m \boxed{F^F_{\leqslant:} \preceq_m F^D_{\leqslant:}} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

The final variant incorporates eager substitution in the quantifier rule:

$$\frac{\vdash^i_F \tau[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w] \leqslant: \sigma[\phi_0/\alpha_0, \ldots, \phi_w/\alpha_w]}{\vdash^{\mathsf{S}i}_F \forall_{\alpha_0 \leqslant: \top, \ldots, \alpha_w \leqslant: \top}.\ \overline{\sigma} \leqslant: \forall_{\alpha_0 \leqslant: \phi_0, \ldots, \alpha_w \leqslant: \phi_w}.\ \overline{\tau}}\ \mathsf{FAllFlip}$$

Theorem 4

$(\exists i.\ \vdash^i_F \sigma \leqslant: \tau) \iff (\exists j.\ \vdash^j_D \sigma \leqslant: \tau)$

Proof.

($\Rightarrow$) By induction on the derivation.
($\Leftarrow$) The new quantifier rule skips all the instances of the variable rule, we use complete induction on the height of the derivation again. $\square$

System $F^F_{\leqslant:}$ (flattened)

$$\boxed{\mathsf{RM} \preceq_m F^F_{\leqslant:}} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

Finally, we can show the reduction from RM halting.

Theorem 5

R halts $\iff \exists i.\ \vdash^i_F \sigma\leqslant:\mathcal{T}(R)$

System $F^F_{\leqslant:}$ (flattened)

$$\boxed{\mathsf{RM} \preceq_m F^F_{\leqslant:}} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

Finally, we can show the reduction from RM halting.

Theorem 5

R halts $\iff \exists i.\ \vdash^i_F \sigma \leqslant: \mathcal{T}(R)$

Proof.

($\Rightarrow$) By induction on the trace.
($\Leftarrow$) One step of the machine corresponds to two applications of the quantifier rule, once again we do complete induction on the height of the derivation. $\square$

Summary

$$\underbrace{\mathsf{2CM} \preceq_m \mathsf{RM}}_{\text{mechanized by first talk}} \quad \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

Summary

$$\underbrace{\mathsf{2CM} \preceq_m \mathsf{RM}}_{\text{mechanized by first talk}} \quad \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

- Syntax directed subtyping, better suited to analyze derivations.

Summary

$$\underbrace{\mathsf{2CM} \preceq_m \mathsf{RM}}_{\text{mechanized by first talk}} \qquad \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

- Syntax directed subtyping, better suited to analyze derivations.
- Polarized syntax enables eager substitution.

Summary

$$\underbrace{\mathsf{2CM} \preceq_m \mathsf{RM}}_{\text{mechanized by first talk}} \quad \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

- Syntax directed subtyping, better suited to analyze derivations.
- Polarized syntax enables eager substitution.
- Well-scoped polyadic syntax profiting from Autosubst2 features.

Summary

$$\underbrace{\mathsf{2CM} \preceq_m \mathsf{RM}}_{\text{mechanized by first talk}} \quad \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

- Syntax directed subtyping, better suited to analyze derivations.
- Polarized syntax enables eager substitution.
- Well-scoped polyadic syntax profiting from Autosubst2 features.
- Induction on height of derivations is required in most proofs.

Summary

$$\underbrace{\mathsf{2CM} \preceq_m \mathsf{RM}}_{\text{mechanized by first talk}} \quad \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

- Syntax directed subtyping, better suited to analyze derivations.
- Polarized syntax enables eager substitution.
- Well-scoped polyadic syntax profiting from Autosubst2 features.
- Induction on height of derivations is required in most proofs.
- Construction of derivations corresponds to a deterministic state transformation.

Summary of mechanization

	LOC	
	Spec.	Proof
Shared facts	500	400
Autosubst2 syntax:		
unscoped	130	20
well-scoped	200	150
Reductions:		
$F^N_{\leqslant:} \preceq_m F_{\leqslant:}$	30	60
$F^D_{\leqslant:} \preceq_m F^N_{\leqslant:}$	150	200
$F^F_{\leqslant:} \preceq_m F^D_{\leqslant:}$	50	100
$\mathsf{RM} \preceq_m F^F_{\leqslant:}$	80	120
$\mathsf{CM2} \preceq_m \mathsf{RM}$	100	50
Total	2340	

Future work

After completing the missing lemma there are further undecidability results that reuse parts of the proof:

- Wehr and Thiemann [2009] reduce $F^{D}_{\leqslant:}$ subtyping to subtyping existential types with upper $(\exists x \leqslant: \tau.\ \sigma)$ and lower $(\exists \tau \leqslant: x.\ \sigma)$ bounds.
  Involves the polarized syntax, might be challenging to mechanize.

Future work

After completing the missing lemma there are further undecidability results that reuse parts of the proof:

- Wehr and Thiemann [2009] reduce $F^{D}_{\leqslant:}$ subtyping to subtyping existential types with upper $(\exists x \leqslant: \tau.\ \sigma)$ and lower $(\exists \tau \leqslant: x.\ \sigma)$ bounds.
  Involves the polarized syntax, might be challenging to mechanize.

Alternatively, the following can readily be mechanized:

- Hu and Lhoták [2020] reduce $F^{N}_{\leqslant:}$ subtyping to subtyping Dependent-Object types (the core calculus of Scala).
  Already mechanized in Agda, porting to Coq should be straightforward.

Future work

Additionally, it would be nice to have the decidability of some variants. There are two approaches:

- Restricting the All rule, e.g. kernel $F_{\leqslant:}$, so the rules induce a terminating algorithm.
- Generalizing bounded quantification, e.g. Maclean and Luo [2021] use Subtype Universes.

$$\mathsf{RM} \preceq_m F^F_{\leqslant:} \preceq_m F^D_{\leqslant:} \preceq_m F^N_{\leqslant:} \preceq_m F_{\leqslant:}$$

$F^N_{\leqslant:}$ Syntax directed.

$F^D_{\leqslant:}$ Deterministic, polarized syntax.

$F^F_{\leqslant:}$ Eager substitution.

	LOC
Autosubst2 syntax	500
Shared facts	900
Reductions	940
Total	2340

Bibliography

Luca Cardelli and Peter Wegner. On understanding types, data abstraction, and polymorphism, 1985.

Giorgio Ghelli. *Proof Theoretic Studies about a minimal type system integrating inclusion and parametric polymorphism*. Università di Pisa. Dipartimento di Informatica, 1990.

Giorgio Ghelli. Divergence of fsub type checking, 1992.

Benjamin C. Pierce. Bounded quantification is undecidable. *Information and Computation*, pages 131–165, 1994.

Stefan Wehr and Peter Thiemann. On the decidability of subtyping with bounded existential types. *Programming Languages and Systems*, 2009.

Jason Z. S. Hu and Ondrej Lhoták. Undecidability of dsub and its decidable fragments. *Proceedings of the ACM on Programming Languages*, 4, 2020. doi: 10.1145/3371077. URL `https://doi.org/10.1145/3371077`.

Harry Maclean and Zhaohui Luo. Subtype universes. volume 188, page 1–16, 2021. ISBN 978-3-95977-182-5. doi: 10.4230/LIPIcs.TYPES.2020.9. URL `https://drops.dagstuhl.de/opus/volltexte/2021/13888`.

Apendix: missing lemma

A translation of renamings is needed, in particular we require:

$$[\![\tau\langle\uparrow\rangle]\!] = [\![\tau]\!]\langle\uparrow^w\rangle$$

The quantifier case is problematic, as translating polyadic binders to regular ones intruduces shiftings:

$$[\![\forall_{\alpha_0 \leqslant: \phi_0, \alpha_1 \leqslant: \phi_1, \ldots, \alpha_w \leqslant: \phi_w}.\ \overline{\tau}]\!]$$
$$= \forall_{[\![\alpha_0]\!] \leqslant: [\![\phi_0]\!]} \forall_{[\![\alpha_1]\!] \leqslant: [\![\phi_1]\!]\langle\uparrow\rangle} \cdots \forall_{[\![\alpha_w]\!] \leqslant: [\![\phi_w]\!]\langle\uparrow^w\rangle}.\ \overline{[\![\tau]\!]}$$