# Towards mechanized undecidability of subtyping in System F

Roberto Álvarez
Advisor: Yannick Forster

Saarland University
Programming Systems Lab

24.06.2021

# System $F$

Terms, types and contexts:

$$t, u ::= x \mid \lambda x : \tau.\, t \mid t\, u \mid \Lambda\alpha.\, t \mid t\, \tau$$
$$\sigma, \tau ::= \alpha \mid \sigma \to \tau \mid \forall\alpha.\, \tau$$
$$\Gamma ::= [] \mid \Gamma,\, x : \tau \mid \Gamma, \alpha \text{ type}$$

Typing rules:

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}$$

$$\frac{\Gamma, x : \sigma \vdash t : \tau}{\Gamma \vdash \lambda x : \sigma.t \;:\; \sigma \to \tau} \qquad \frac{\Gamma \vdash t : \sigma \to \tau \qquad \Gamma \vdash u : \sigma}{\Gamma \vdash t\, u \;:\; \tau}$$

$$\frac{\Gamma, \alpha \text{ type} \vdash t : \tau}{\Gamma \vdash \Lambda\alpha.t \;:\; \forall\alpha.\tau} \qquad \frac{\Gamma \vdash t : \forall\alpha.\tau \qquad \Gamma \vdash \sigma \text{ type}}{\Gamma \vdash t\, \sigma \;:\; \tau[\sigma/\alpha]}$$

# Subtyping

Reflexive and transitive relation on types $\sigma \leqslant: \tau$

$$\frac{\Gamma \vdash t : \sigma \qquad \Gamma \vdash \sigma \leqslant: \tau}{\Gamma \vdash t : \tau}$$

Any term of type $\sigma$ may be used where a term of type $\tau$ is expected:

$$\begin{aligned} \mathbb{N} &\leqslant: \mathbb{R} \\ \{x : X, y : Y\} &\leqslant: \{x : X\} \\ \tau &\leqslant: \top \end{aligned}$$

# System $F_{\leqslant:}$

Contexts also contain type bounds: $\Gamma ::= \cdots \mid \Gamma,\ \alpha \leqslant: \tau$

Bounded quantification replaces universal quantification

$$\frac{\Gamma, \alpha \leqslant: \sigma \vdash t : \tau}{\Gamma \vdash \Lambda \alpha \leqslant: \sigma.\ t\ :\ \forall \alpha \leqslant: \sigma.\ \tau}$$

$$\forall \alpha . \tau := \forall \alpha \leqslant: \top . \tau$$

Bounded quantification is contravariant in the bounds:

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

$\sigma_2$ gets rebounded with $\tau_1$, this allows to encode ”registers” and numbers.

# $F_{\leqslant:}$ subtyping

For arbitrary $\Gamma$, $\sigma$ and $\tau$, is there a derivation of $\Gamma \vdash \sigma \leqslant: \tau$?

[Pierce, 1994] gives a proof of undecidability:

## Theorem

*$F_{\leqslant:}$ subtyping is undecidable*

## Proof.

By a chain of many-one reductions

2CM halting $\leq_m$ RM halting $\leq_m F^F_{\leqslant:}$ subtyping
$\leq_m F^D_{\leqslant:}$ subtyping
$\leq_m F^N_{\leqslant:}$ subtyping
$\leq_m F_{\leqslant:}$ subtyping $\square$

# Variants of System $F_{\leqslant:}$

Each variant refines the rules further to simplify the translation from rowing machines.

- $F^{N}_{\leqslant:}$ Syntax directed subtyping
- $F^{D}_{\leqslant:}$ Deterministic subtyping; polarized syntax
- $F^{F}_{\leqslant:}$ Eager substitution; empty context

The full system is a conservative extension of each variant.

## Lemma

*If $L_1$ is a conservative extension of $L_2$ then there is a many-one reduction*

$$l \in L_2 \leq_m l \in L_1$$

# Rowing machines

A closed $n$-tuple of rows $\langle \rho_1 \ldots \rho_n \rangle$ each row is of the form

$$\rho ::= \alpha_i \mid \langle \alpha_1 \ldots \alpha_n;\ \rho_1 \ldots \rho_n \rangle \mid \texttt{Halt} \qquad i \in \{1 \ldots n\}$$

The step function for $\rho_1 = \langle \alpha_1 \ldots \alpha_n;\ \rho'_1 \ldots \rho'_n \rangle$

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

For example

$$\begin{aligned} & \langle \langle \alpha_1, \alpha_2, \alpha_3;\ \alpha_1, \alpha_3, \alpha_2 \rangle, A, B \rangle \\ \mapsto & \langle \langle \alpha_1, \alpha_2, \alpha_3;\ \alpha_1, \alpha_3, \alpha_2 \rangle, B, A \rangle \\ \mapsto & \langle \langle \alpha_1, \alpha_2, \alpha_3;\ \alpha_1, \alpha_3, \alpha_2 \rangle, A, B \rangle \\ \mapsto & \ldots \end{aligned}$$

# Rowing machines

$\mathcal{F}$ encodes rowing machines as $F^F_{\leqslant:}$ statements and rows as types.

The contents of the registers are *stored* as bounds and the next instruction as the body:

$$\mathcal{F}(\langle \rho_1 \dots \rho_n \rangle) := \sigma \leqslant: \dots \forall \beta_1 \leqslant: \mathcal{F}(\rho_1) \dots \forall \beta_n \leqslant: \mathcal{F}(\rho_n).\ (\dots \mathcal{F}(\rho_1) \dots)$$

The type $\sigma$ and the rest of the right hand side reorder the bounds to yield a statement of the same form with the bounds substituted in the body.

If $R \to_R R'$ and there is a derivation of $\mathcal{F}(R')$ then there is a derivation of $\mathcal{F}(R)$.

## Lemma

*The rowing machine $R$ halts iff there is a derivation of $\mathcal{F}(R)$.*

# 2-counter machines

[Pierce, 1994] defines 2-counter machines as a program counter and two registers $(PC, A, B)$, instructions are of the form:

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

[Dudenhefner, 2021] gives a slightly different presentation of counter machines, however Pierce's machines are more easily translated to rowing machines.

# 2-counter machines

$\mathcal{R}$ encodes 2-counter machines, instructions and natural numbers as rowing machines.

When used as an instruction the encoding of a number loads the next instruction and its predecessor (itself in case of 0) in the appropriate registers.

The ability of rowing machines to encode numbers, successors and predecessors motivates the use of counter machines.

## Lemma

*The 2-counter machine $M$ halts iff $\mathcal{R}(M)$ halts.*

# Mechanized proofs

Completed:

- Undecidability of Pierce's 2CM halting.
- Undecidability of RM halting.

In development:

- RM halting many-one reduces to $F^{F}_{\leqslant:}$ subtyping.
- The full system is indeed conservative extensions of the variants.

# Bibliography


Dudenhefner, A. (2021).

Constructive many-one reduction from the halting problem to semi-unification.

Pierce, B. C. (1994).

Bounded quantification is undecidable.

*Information and Computation*, pages 131–165.

Pierce, B. C. (2002).

*Types and programming languages*.

MIT Press.

# System $F_{\leqslant:}$

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

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

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

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

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

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

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

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

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

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

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

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

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

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