Preciseness of Subtyping on Intersection and Union Types

Silvia Ghilezan

University of Novi Sad
Serbia

Université Paris Diderot, 21 April 2016

Subtyping

Joint work with Mariangiola Dezani-Ciancaglini

M. Dezani-Ciancaglini and SG.
Preciseness of subtyping on intersection and union types.
In *RTA-TLCA 2014*, volume 8560 of *LNCS*, pages 194–207 (2014).

Subtyping

Subtyping is a binary relation $\leq$ (preorder) on the set of `Types`

$$\sigma \leq \tau$$

Subsumption rule in the type inference system

$$\frac{M : \sigma \quad \sigma \leq \tau}{M : \tau}$$

- λ-calculi, concurrent calculi
- programming languages

1. Soundness and completeness
2. Concurrent λ-calculus
3. Preciseness Results
4. Conclusion

1 Soundness and completeness

2 Concurrent λ-calculus

3 Preciseness Results

4 Conclusion

Preciseness of subtyping

Preciseness
- Soundness
- Completeness

Two aspects:
- Denotational preciseness
- Operational preciseness

Denotational Preciseness of Subtyping

$[\![\sigma]\!]$ is a set interpreting type σ

denotational soundness: $\sigma \leq \tau$ implies $[\![\sigma]\!] \subseteq [\![\tau]\!]$

denotational completeness: $[\![\sigma]\!] \subseteq [\![\tau]\!]$ implies $\sigma \leq \tau$

denotational preciseness: $\sigma \leq \tau$ iff $[\![\sigma]\!] \subseteq [\![\tau]\!]$


H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini.
A Filter Lambda Model and the Completeness of Type Assignment.
Journal of Symbolic Logic, 48(4):931–940, 1983.


J. Vouillon.
Subtyping Union Types.
In *CSL*, volume 3210 of *LNCS*, pages 415–429, 2004.

Operational Soundness of Subtyping

If $\sigma \leq \tau$, **then** each context
- that is safe when filled with a term of type τ is also safe when filled with a term of type σ

$$\forall C[\,]\ (\forall M : \tau\ C[M] \not\to^* \texttt{error} \implies \forall N : \sigma\ C[N] \not\to^* \texttt{error})$$

Example. $\texttt{nat} \leq \texttt{int}$ $C[-5]$ converges, then $C[2]$ converges

Subsumption rule in the type system

$$\frac{M : \sigma \quad \sigma \leq \tau}{M : \tau}$$

Operational soundness of subtyping follows from subject reduction of the type system

Operational Completeness of Subtyping

Converse:
If each context that is safe when filled with a term of type τ is also safe when filled with a term of type σ, **then** $\sigma \leq \tau$

Instead:
If $\sigma \not\leq \tau$, **then** there is a context

- that is safe when filled with an arbitrary term of type τ, and
- gives an error when filled with a suitable term of type σ

$$\exists C_0[\,](\forall M : \tau.\ C_0[M] \not\to^* \texttt{error} \bigwedge \exists N_0 : \sigma. C_0[N_0] \to^* \texttt{error})$$

1. Soundness and completeness
2. Concurrent λ-calculus
3. Preciseness Results
4. Conclusion

Operational Preciseness of Subtyping

soundness and completeness

$\sigma \leq \tau$ **iff** for each context

- that is safe when filled with a term of type τ is also safe when filled with a term of type σ

$$\forall C[\,](\forall M : \tau.\ C[M] \not\to^* \texttt{error} \implies \forall N : \sigma.\ C[N] \not\to^* \texttt{error})$$

$\sigma \leq \tau$ **iff** there is no context

- that is safe when filled with an arbitrary term of type τ and
- gives an error when filled with a suitable term of type σ

$$\neg\exists C_0[\,](\forall M : \tau.\ C_0[M] \not\to^* \texttt{error} \bigwedge \exists N_0 : \sigma C_0[M_0] \to^* \texttt{error}\ )$$

J. Blackburn, I. Hernandez, J. Ligatti, and M. Nachtigal.
Completely subtyping iso-recursive types.
Technical Report, University of South Florida, 2014.

Concurrent λ-calculus - Syntax

M. Dezani-Ciancaglini, U. de'Liguoro, and A. Piperno.
A Filter Model for Concurrent Lambda-Calculus.
SIAM Journal on Computing 27(5):1376–1419, 1998.

$$M ::= x \mid v \mid (\lambda x.M) \mid (\lambda v.M) \mid (MM) \mid (M + M) \mid (M \| M)$$

1. call-by-name and call-by-value variables
2. internal choice
3. parallel operator

$W ::= v \mid \lambda x.M \mid \lambda v.M \mid W \| W$
$V ::= W \mid V \| M \mid M \| V$

TVal total values:
Val values

Reduction rules

$$(+_L)\ M+N \longrightarrow M \qquad (+_R)\ M+N \longrightarrow N$$

$$(\|_{app})\ (M\|N)L \longrightarrow ML\|NL \qquad (\|_s)\ \frac{M \longrightarrow M' \quad N \longrightarrow N'}{M\|N \longrightarrow M'\|N'}$$

$$(\|_a)\ \frac{M \longrightarrow M' \quad W \in \mathrm{TVal}}{M\|W \longrightarrow M'\|W,\ W\|M \longrightarrow W\|M'}$$

$$(\beta)\ (\lambda x.M)N \longrightarrow M[N/x] \qquad (\beta_v)\ \frac{W \in \mathrm{TVal}}{(\lambda v.M)W \longrightarrow M[W/v]}$$

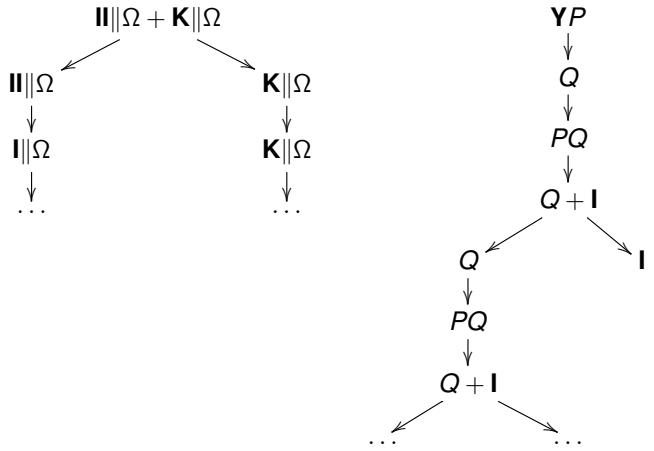
$$(\beta_v\|)\ \frac{V \longrightarrow V' \quad V \in \mathrm{Val}}{(\lambda v.M)V \longrightarrow M[V/v]\|(\lambda v.M)V'}$$

$$(\mu_v)\ \frac{N \longrightarrow N' \quad N \notin \mathrm{Val}}{(\lambda v.M)N \longrightarrow (\lambda v.M)N'} \qquad (\nu)\ \frac{M \longrightarrow M' \quad M \notin \mathrm{Val} \bigcup \mathrm{Par}}{MN \longrightarrow M'N}$$

TVal *total values*: $W ::= v \mid \lambda x.M \mid \lambda v.M \mid W\|W$
Val *values* $V ::= W \mid V\|M \mid M\|V$
$\mathrm{Par} = \{M\|N\}$

Convergence

reduction tree $P = \lambda x.(x + \mathbf{I})$ $Q = (\lambda x.P(xx))(\lambda x.P(xx))$
Bar is a subset of nodes of the reduction tree such that each maximal path intersects the bar at exactly one node
a term converges if there is a bar of values in its reduction tree



Types and Subtyping

Type: $\sigma ::= \omega \mid \sigma \to \sigma \mid \sigma \wedge \sigma \mid \sigma \vee \sigma$

$\sigma \leq \tau$ is the smallest pre-order on types such that

1. $\langle \mathrm{Type}, \leq \rangle$ is a distributive lattice, in which $\wedge$ is the meet, $\vee$ is the join and ω is the top;
2. the arrow satisfies
   1. $\sigma \to \omega \leq \omega \to \omega$;
   2. $(\sigma \to \rho) \wedge (\sigma \to \tau) \leq \sigma \to \rho \wedge \tau$;
   3. $\sigma \geq \sigma', \tau \leq \tau' \Rightarrow \sigma \to \tau \leq \sigma' \to \tau'$.

CType: a type σ is coprime if $\sigma \leq \tau \vee \rho$ implies $\sigma \leq \tau$ or $\sigma \leq \rho$

Each type is equal to an union of coprime types.

Typing Rules

A basis Γ maps

1. call-by-name variables to types (ω by default) and
2. call-by-value variables to coprime types ($\omega \to \omega$ by default)

$$(\mathrm{Ax})\ \Gamma \vdash \alpha : \Gamma(\alpha) \qquad (\omega)\ \Gamma \vdash M : \omega$$

$$(\to \mathrm{I}_n)\ \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau}$$

$$(\to \mathrm{I}_v)\ \frac{\Gamma, v : \sigma_i \vdash M : \tau \quad \sigma = \bigvee_{i \in I} \sigma_i \quad \sigma_i \in \mathrm{CType}\ i \in I}{\Gamma \vdash \lambda v.M : \sigma \to \tau}$$

$$(\to \mathrm{E})\ \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

$$(\wedge \mathrm{I})\ \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \wedge \tau} \qquad (\leq)\ \frac{\Gamma \vdash M : \sigma \quad \sigma \leq \tau}{\Gamma \vdash M : \tau}$$

$$(+\mathrm{I})\ \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau}{\Gamma \vdash M + N : \sigma \vee \tau} \qquad (\|\,\mathrm{I})\ \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau}{\Gamma \vdash M\|N : \sigma \wedge \tau}$$

Characterisation of Convergence

Each type is either a subtype of $\omega \to \omega$ or it is equivalent to ω.

Theorem (Type preservation)
The type system enjoys subject reduction.

Theorem
A closed term is convergent iff it has type $\omega \to \omega$.

Corollary
A closed term is divergent iff it has only types equivalent to ω.

Unsoundness of $(\sigma \to \rho) \wedge (\tau \to \rho) \leq \sigma \vee \tau \to \rho$

$\sigma = \rho \to \omega \to \rho \qquad \tau = \omega \to \rho \to \rho \qquad \rho = \omega \to \omega$

$\vdash \lambda x.x\mathbf{I}\Omega \| \lambda x.x\Omega\mathbf{I} : (\sigma \to \rho) \to (\tau \to \rho)$ and $\vdash \mathbf{K} + \mathbf{O} : \sigma \vee \tau$

$\vdash (\lambda x.x\mathbf{I}\Omega \| \lambda x.x\Omega\mathbf{I})(\mathbf{K} + \mathbf{O}) : \rho \ \ (= \omega \to \omega)$

$$(\lambda x.x\mathbf{I}\Omega \| \lambda x.\Omega\mathbf{I})(\mathbf{K} + \mathbf{O}) \longrightarrow (\mathbf{K} + \mathbf{O})\mathbf{I}\Omega \| (\mathbf{K} + \mathbf{O})\Omega\mathbf{I}) \longrightarrow \mathbf{O}\mathbf{I}\Omega \| \mathbf{K}\Omega\mathbf{I} \longrightarrow \Omega \| \Omega$$

$\Omega \| \Omega$ diverges $\qquad \not\vdash \Omega \| \Omega : \omega \to \omega \qquad$ subject reduction fails!

1. Soundness and completeness
2. Concurrent λ-calculus
3. Preciseness Results
4. Conclusion

Preciseness for the Concurrent λ-calculus

The subtyping $\leq$ is denotationally precise when

$$\sigma \leq \tau \ \text{ if and only if } \ [\![\sigma]\!] \subseteq [\![\tau]\!]$$

The subtyping $\leq$ is operationally precise when

$\sigma \leq \tau$ if and only if
there is no closed terms M_0 such that $M_0 P$ converges for all closed terms $P : \tau$ and for some $N_0 : \sigma$ the term $M_0 N_0$ diverges

$$\neg \exists M_0[\,](\forall P : \tau.\ M_0 P \text{ conveges} \bigwedge \exists N_0 : \sigma.\ M_0 N_0 \text{ diverges})$$

Operational preciseness for the Concurrent λ-calculus

key terms

$\mathsf{R}_\omega = \Omega$; $\quad \mathsf{T}_\omega = \lambda xy.y$;
$\mathsf{R}_{\sigma\to\tau} = \lambda x.(\mathsf{T}_\sigma x)\,\mathsf{R}_\tau$; $\quad \mathsf{T}_{\sigma\to\tau} = \lambda v.\mathsf{T}_\tau(v\,\mathsf{R}_\sigma)$;
$\mathsf{R}_{\sigma\wedge\tau} = \mathsf{R}_\sigma \| \mathsf{R}_\tau$; $\quad \mathsf{T}_{\sigma\wedge\tau} = \lambda x.(\mathsf{T}_\sigma\, x + \mathsf{T}_\tau\, x)$;
$\mathsf{R}_{\sigma\vee\tau} = \mathsf{R}_\sigma + \mathsf{R}_\tau$. $\quad \mathsf{T}_{\sigma\vee\tau} = \lambda v.(\mathsf{T}_\sigma\, v \,\|\, \mathsf{T}_\tau\, v)$ where $\sigma\vee\tau \neq \omega$.

characteristic term

R_σ is the "worst" (with respect to convergence) term of type σ
R_σ has type τ iff $\sigma \leq \tau$

test term

$\mathsf{T}_\sigma M$ (M closed) converges (to $\mathbf{I}$) iff M has type σ
T_σ has type $\tau\to\rho\to\rho$ iff $\tau \leq \sigma$

Theorem
The subtyping $\leq$ is denotationally precise for the concurrent λ-calculus.

$$[\![\sigma]\!] = \{M \mid \vdash M : \sigma\}$$

Theorem
The subtyping $\leq$ is operationally precise for the concurrent λ-calculus.

1. Soundness and completeness
2. Concurrent λ-calculus
3. Preciseness Results
4. Conclusion

Preciseness for Pure λ-Calculus

Operational completeness requires that all empty (i.e. not inhabited) types are less than all inhabited types

Inhabitation is undecidable for polymorphic types and for intersection types

A complete subtyping on polymorphic types or on intersection types for the pure λ-calculus must be undecidable

This makes unfeasible an operationally complete subtyping for the pure λ-calculus, both in case of polymorphic types and intersection and union types

The terms of the concurrent λ-calculus inhabit all types

Open problem: to study the extensions of λ-calculus enjoying operational preciseness for the decidable subtyping between polymorphic types

Related work

J. Blackburn, I. Hernandez, J. Ligatti, and M. Nachtigal.
Completely subtyping iso-recursive types.
Technical Report, University of South Florida, 2014.

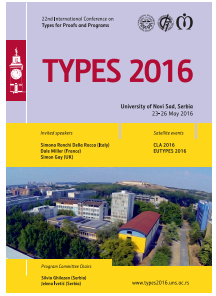
T. Chen, M. Dezani-Ciancaglini, and N. Yoshida.
On the Preciseness of Subtyping in Session Types.
In *PPDP 2014*, 135–146, 2014.

M. Dezani-Ciancaglini, SG, S. Jaksic, J. Pantovic and N. Yoshida.
Precise subtyping for synchronous multiparty sessions.
In *PLACES 2015*, EPTCS 203:29–43, 2016.

M. Dezani-Ciancaglini, SG, S. Jaksic, J. Pantovic and N. Yoshida.
Denotational and Operational Preciseness of Subtyping: A Roadmap.
In *Theory and Practice of Formal Methods 2016*, LNCS 9660: 155–172, 2016.

TYPES 2016, May 23-27, 2016, Novi Sad

- **TYPES 2016** - 22nd International Conference on **Types for Proofs and Programs**
- Affiliated events: CLA 2016, EUTYPES 2016
- **Novi Sad, Serbia**
- **May 23- 27, 2016**
- http://www.types2016.uns.ac.rs/



Thanks