Negations
in Refinement Type Systems

T. Tsukada (U. Tokyo)

22nd Sep. 2016

IMS, Singapore

This Talk

About refinement intersection type systems that refute judgements of other type systems.

$$\nvdash M : \tau$$

$$\iff \quad \Vdash M : \neg\tau$$

Background

Refinement intersection type systems are the basis for

- model checkers of higher-order model checking (cf. [Kobayashi 09] [Broadbent&Kobayashi 11] [Ramsay+ 14]),
- software model-checker for higher-order programs (cf. MoCHi [Kobayashi+ 11]).

In those type systems,

- a derivation gives a witness of derivability,
- but nothing witnesses that a given derivation is not derivable.

Motivation

A witness of underivability would be useful for

- a compact representation of an error trace
- an efficient model-checker in collaboration with the affirmative system
  - cf. [Ramsay+ 14] [Godefroid+ 10]
- development of a type system proving safety
  - In some cases (e.g. [T&Kobayashi 14]), a type system proving failure is easier to be developed.

Contribution

Development of type systems refuting derivability in some type systems such as

- a basic type system for the λ-calculus
- a type system for call-by-value reachability

Theoretical study of the development

Outline

- Reviewing a refinement intersection type system for higher-order model checking
- Negative type system
- Extensions
- Discussions

Target language: CbN $\lambda^{\to}$-calculus

A simply typed calculus equipped with $\beta\eta$-equivalence.

Kinds (i.e. simple types):

$$A, B ::= o \mid A \to A$$

Terms:

$$M, N ::= x \mid \lambda x^A.M \mid M\,M$$

Typing rules:

$$\frac{(x :: A) \in \Delta}{\Delta \vdash x :: A} \qquad \frac{\Delta, x :: A \vdash M :: B}{\Delta \vdash \lambda x^A.M :: A \to B}$$

$$\frac{\Delta \vdash M :: A \to B \qquad \Delta \vdash N :: A}{\Delta \vdash M\,N :: B}$$

Refinement types

Types are parameterised by kinds and ground type sets:

$$\mathrm{Ty}_Q(o) := Q$$
$$\mathrm{Ty}_Q(A \to B) := \mathcal{P}(\mathrm{Ty}_Q(A)) \times \mathrm{Ty}_Q(B)$$

We use the following syntax for types:

$$\tau, \sigma ::= q \mid X \to \tau$$
$$X, Y \in \mathcal{P}(\mathrm{Ty}_Q(A))$$

Alternative definition

Let A be a kind.

The set $\mathrm{Ty}_Q(A)$ of types that refines A is given by

$$\mathrm{Ty}_Q(A) = \{\, \tau \mid \tau :: A \,\}$$

where is the refinement relation:

$$\frac{q \in Q}{q :: o} \qquad \frac{\forall \sigma \in X. \sigma :: A \qquad \tau :: B}{(X \to \tau) :: A \to B}$$

Subtyping

The subtyping relation is defined by induction on kinds.

$$\frac{}{q \preceq_o q}$$

$$\frac{X \succeq_{!A} Y \qquad \tau \preceq_B \sigma}{(X \to \tau) \preceq_{A \to B} (Y \to \sigma)}$$

$$\frac{\forall \sigma \in Y. \exists \tau \in X. \tau \preceq_A \sigma}{X \preceq_{!A} Y}$$

Type Environments

A (finite) map from variables to sets of types
(or intersection types).

$$\Gamma ::= x_1 : X_1, \ldots, x_n : X_n \quad (n \geq 0)$$

Typing rules

$$\frac{(x : X) \in \Gamma \qquad \tau \in X \qquad \tau \preceq \sigma}{\Gamma \vdash x : \sigma}$$

$$\frac{\Gamma, x : X \vdash M : \tau}{\Gamma \vdash \lambda x.M : X \to \tau}$$

$$\frac{\Gamma \vdash M : X \to \tau \qquad \Gamma \vdash N : X}{\Gamma \vdash M\,N : \tau}$$

$$\frac{\forall \tau \in X.\, \Gamma \vdash M : \tau}{\Gamma \vdash M : X}$$

Fact: Invariance under $\beta\eta$-equivalence

Suppose that $M =_{\beta\eta} N$. Then

$$\Gamma \vdash M : \tau \Leftrightarrow \Gamma \vdash N : \tau$$

- This fact will not be used in the sequel.

Convention: Subtyping closure

In what follows, sets of types are assumed to be closed under the subtyping relation.

$$\tau \succeq \sigma \in X \Rightarrow \tau \in X$$

The rule for variables becomes simpler.

$$\frac{(x : X) \in \Gamma \qquad \tau \in X}{\Gamma \vdash x : \tau}$$

Outline

- Reviewing a refinement intersection type system
- Negative type system
- Extensions
- Discussions

Negative Types

Negative types are those constructed from the negative ground types $\overline{Q} := \{\, \bar{q} \mid q \in Q \,\}$:

$$\overline{\mathrm{Ty}_Q(A)} := \mathrm{Ty}_{\overline{Q}}(A)$$

$$\bar{\tau}, \bar{\sigma} ::= \bar{q} \mid \bar{X} \to \bar{\tau}$$
$$\bar{X}, \bar{Y} \in u(\mathrm{Ty}_{\bar{Q}}(A))$$

Typing rules are the same as the affirmative system.

Goal and approach

Giving an anti-monotone bijections on prime types

$$\neg_A : \mathrm{Ty}_Q(A) \longrightarrow \overline{\mathrm{Ty}_Q(A)}$$

such that, for every term $M :: A$,

$$\nvdash M : \tau \quad \Leftrightarrow \quad \Vdash M : \neg_A \tau$$

This implies that

$$\neg \exists M.\ (\vdash M : \tau)\ \&\ (\Vdash M : \neg \tau)$$

We shall first study this relation.

(In)consistency cf. [Salvati & Walukiewicz 2011]

(Intuitively) $\tau \in \mathrm{Ty}_Q(A)$ and $\bar{\sigma} \in \overline{\mathrm{Ty}_Q(A)}$ are ***consistent*** if

$$\exists d \in A.\ (d \vDash \tau)\ \&\ (d \vDash \bar{\sigma})$$

and ***inconsistent*** otherwise.

Inference rules

$$\tau \parallel \bar{\sigma} \quad \Leftrightarrow \quad \tau \text{ and } \bar{\sigma} \text{ are consistent}$$

$$\tau \asymp \bar{\sigma} \quad \Leftrightarrow \quad \tau \text{ and } \bar{\sigma} \text{ are inconsistent}$$

$$\frac{q \in Q}{q \asymp \bar{q}} \qquad \frac{\exists \tau \in X. \exists \bar{\sigma} \in \bar{Y}. \tau \asymp \bar{\sigma}}{X \asymp \bar{Y}} \qquad \frac{X \parallel \bar{Y} \qquad \tau \asymp \bar{\sigma}}{(X \to \tau) \asymp (\bar{Y} \to \bar{\sigma})}$$

$$\frac{\neg(\tau \asymp \bar{\sigma})}{\tau \parallel \bar{\sigma}}$$

(In)consistency cf. [Salvati & Walukiewicz 2011]

(Intuitively) $\tau \in \mathrm{Ty}_Q(A)$ and $\bar{\sigma} \in \overline{\mathrm{Ty}_Q(A)}$ are ***consistent*** if

$$\exists d \in A.\ (d \vDash \tau)\ \&\ (d \vDash \bar{\sigma})$$

and ***inconsistent*** otherwise.

Inference rules

$$\begin{aligned} \tau \parallel \bar{\sigma} &\quad \Leftrightarrow \quad \tau \text{ and } \bar{\sigma} \text{ are consistent} \\ \tau \asymp \bar{\sigma} &\quad \Leftrightarrow \quad \tau \text{ and } \bar{\sigma} \text{ are inconsistent} \end{aligned}$$

$$\frac{q \in Q}{q \asymp \bar{q}} \qquad \frac{\exists \tau \in X. \exists \bar{\sigma} \in \bar{Y}. \tau \asymp \bar{\sigma}}{X \asymp \bar{Y}} \qquad \frac{X \parallel \bar{Y} \qquad \tau \asymp \bar{\sigma}}{(X \to \tau) \asymp (\bar{Y} \to \bar{\sigma})}$$

$$\frac{\neg(\tau \asymp \bar{\sigma})}{\tau \parallel \bar{\sigma}}$$

Assume $\exists f.\ (f \vDash X \to \tau)\ \&\ (f \vDash \bar{Y} \to \bar{\sigma})$
Take d s.t. $(d \vDash X)\ \&\ (d \vDash \bar{Y})$
Then $(f(d) \vDash \tau)\ \&\ (f(d) \vDash \bar{\sigma})$, contradiction

Negation is weakest inconsistent type

Recall that

$$\nVdash M : \tau \quad \Leftrightarrow \quad \Vdash M : \neg_A \tau$$

- **[Inconsistent]**

  We have $\tau \asymp \neg\tau$

- **[Weakest]**

  Assume that $\tau \asymp \bar{\sigma}$. Then

$$\begin{aligned} \Vdash M : \bar{\sigma} \quad &\Rightarrow \quad \nVdash M : \tau \\ &\Rightarrow \quad \Vdash M : \neg\tau \end{aligned}$$

Negation is weakest inconsistent type

Recall that

Does it exist?

$$\nvdash M : \tau \quad \Leftrightarrow \quad \Vdash M : \neg_A \tau$$

- **[Inconsistent]**

  We have $\tau \asymp \neg\tau$

- **[Weakest]**

  Assume that $\tau \asymp \bar{\sigma}$. Then

$$\Vdash M : \bar{\sigma} \quad \Rightarrow \quad \nvdash M : \tau$$

$$\Rightarrow \quad \Vdash M : \neg\tau$$

Definition of the negation

Define the two anti-monotone bijections on types

$$\neg_A : \mathrm{Ty}_Q(A) \longrightarrow \overline{\mathrm{Ty}_Q(A)}$$
$$\natural_A : u(\mathrm{Ty}_Q(A)) \longrightarrow u(\overline{\mathrm{Ty}_Q(A)})$$

as follows:

$$\neg_o q := \bar{q}$$
$$\neg_{A\to B}(X \to \tau) := (\natural_A X) \to (\neg_B \tau)$$
$$\natural_A X := \{\, \neg_A \tau \mid \tau \notin X \,\}$$

Definition of the negation

Define t Weakest inconsistent prime type n types

$$\neg_A : \mathrm{Ty}_Q(A) \longrightarrow \overline{\mathrm{Ty}_Q(A)}$$

$$\natural_A : u(\mathrm{Ty}_Q(A)) \longrightarrow u(\overline{\mathrm{Ty}_Q(A)})$$

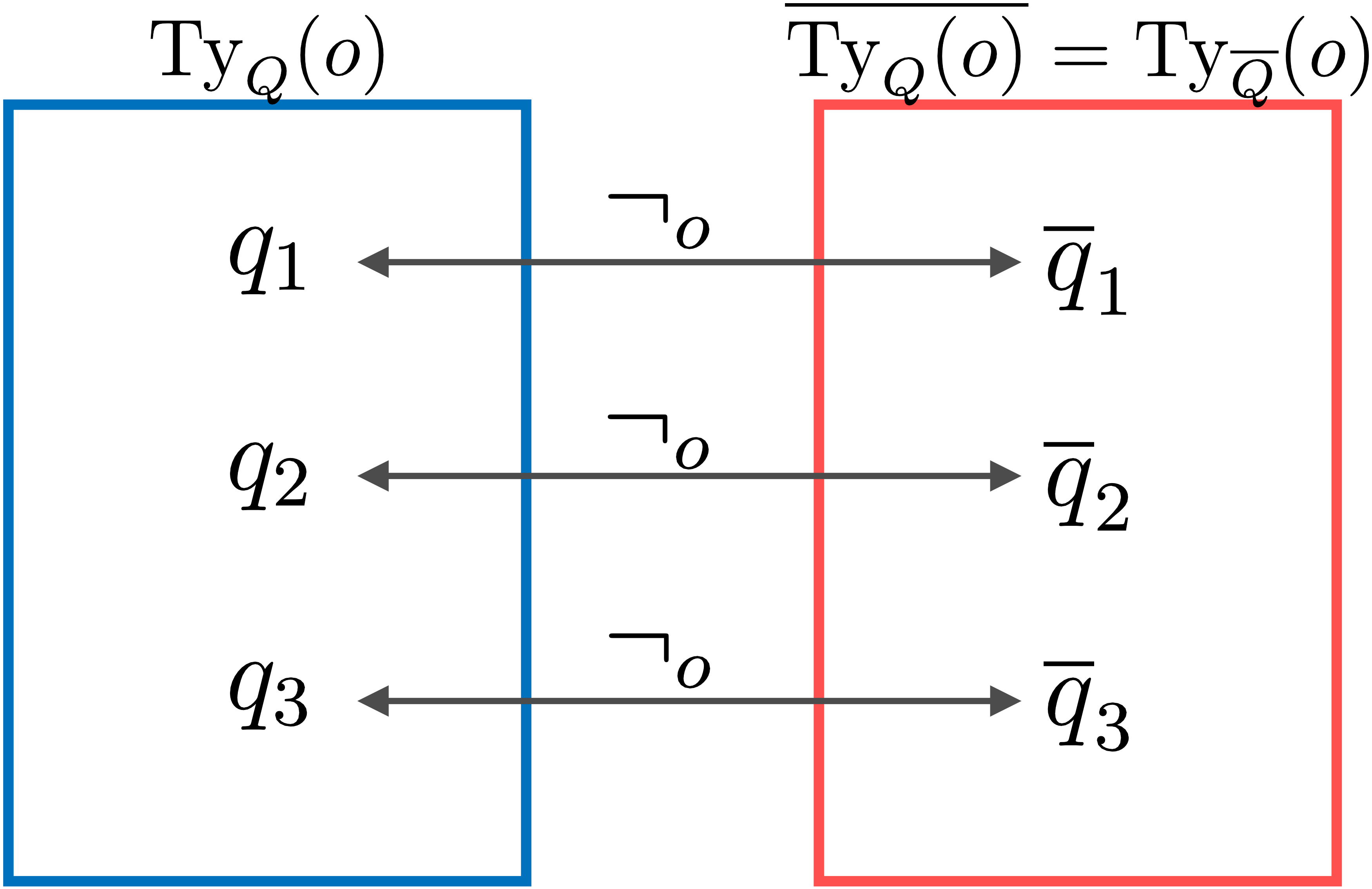
as follows: Strongest consistent intersection type

$$\neg_o q := \bar{q}$$

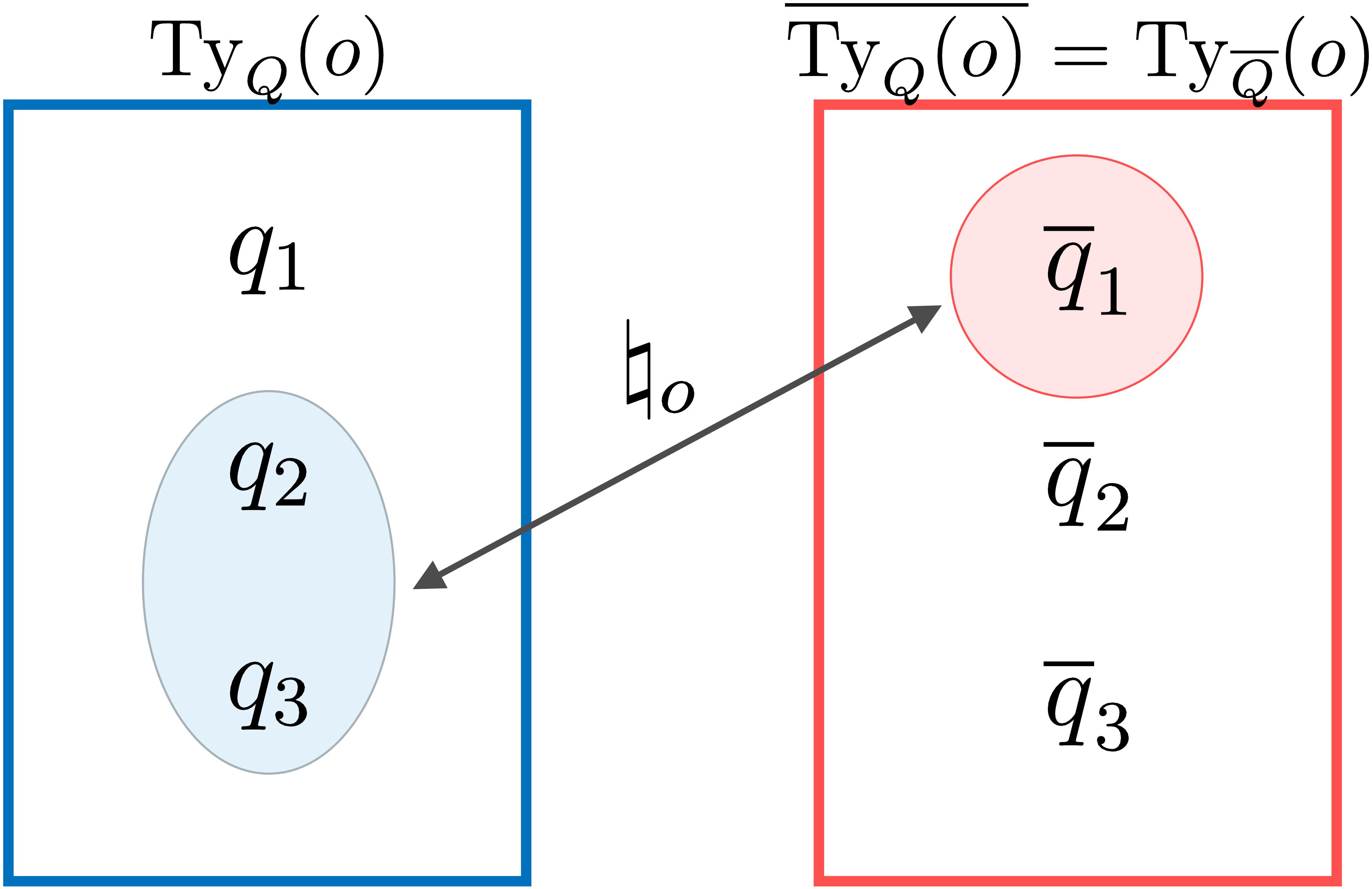
$$\neg_{A \to B}(X \to \tau) := (\natural_A X) \to (\neg_B \tau)$$

$$\natural_A X := \{\, \neg_A \tau \mid \tau \notin X \,\}$$

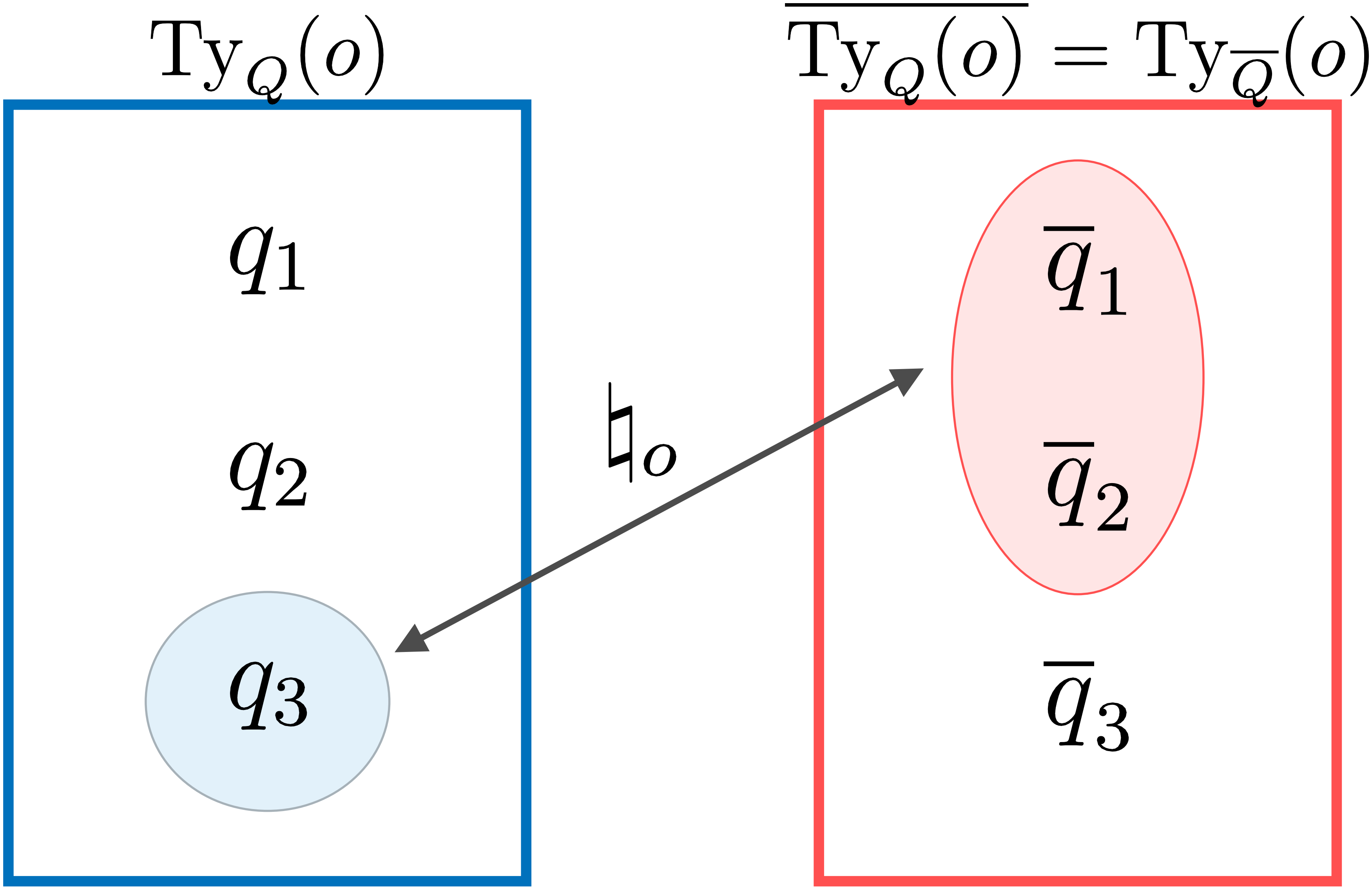
Negation $\neg_A : \mathrm{Ty}_Q(A) \longrightarrow \overline{\mathrm{Ty}_Q(A)}$



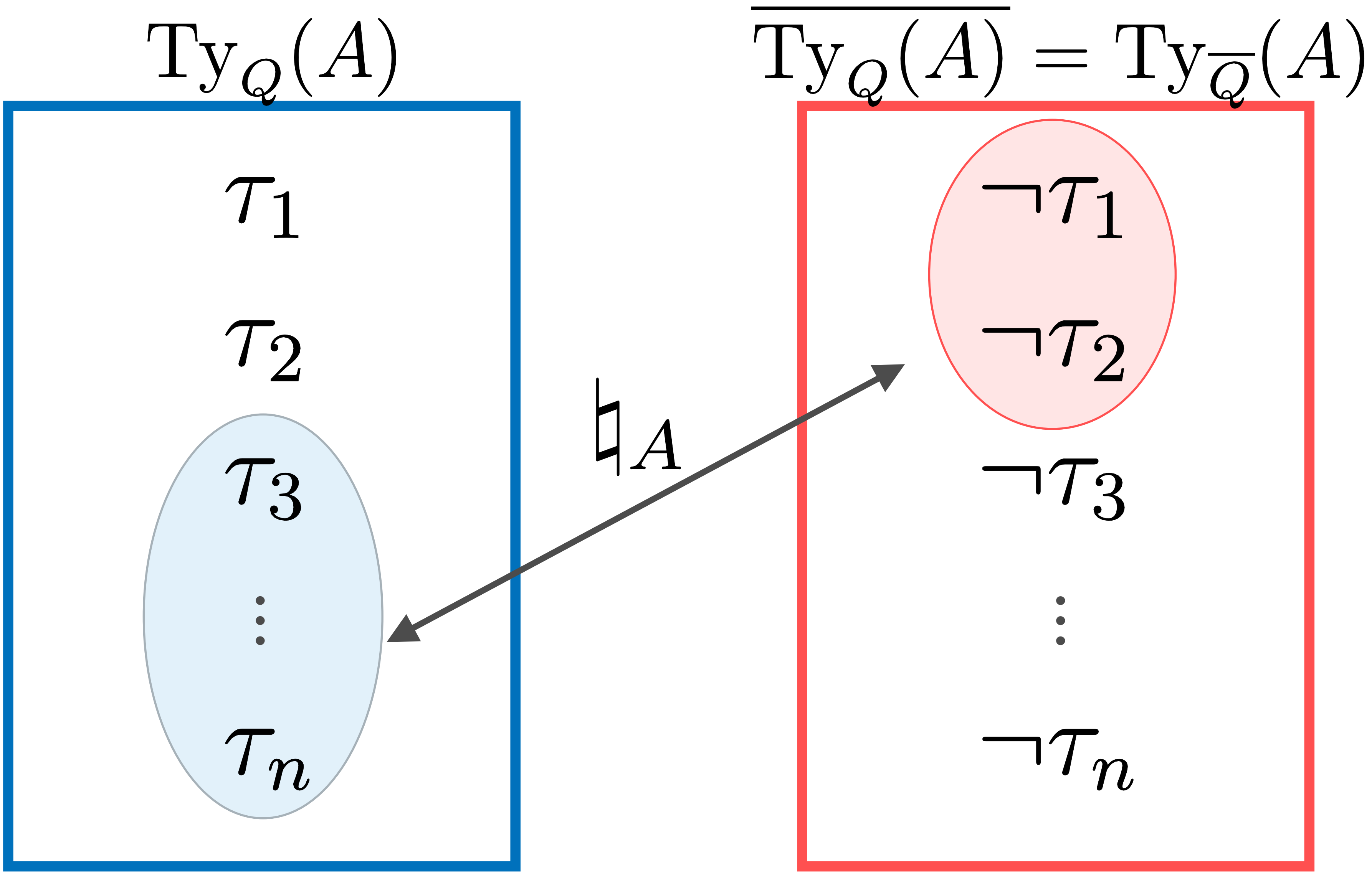
Natural $\quad \natural_A : u(\mathrm{Ty}_Q(A)) \longrightarrow u(\overline{\mathrm{Ty}_Q(A)})$



Natural $\natural_A : u(\mathrm{Ty}_Q(A)) \longrightarrow u(\overline{\mathrm{Ty}_Q(A)})$



Natural $\natural_A : u(\mathrm{Ty}_Q(A)) \longrightarrow u(\overline{\mathrm{Ty}_Q(A)})$



Definition of the negation

Define t Weakest inconsistent prime type n types

$$\neg_A : \mathrm{Ty}_Q(A) \longrightarrow \overline{\mathrm{Ty}_Q(A)}$$

$$\natural_A : u(\mathrm{Ty}_Q(A)) \longrightarrow u(\overline{\mathrm{Ty}_Q(A)})$$

Strongest consistent intersection type

as follows:

$$\neg_o q := \bar{q}$$

$$\neg_{A \to B}(X \to \tau) := (\natural_A X) \to (\neg_B \tau)$$

$$\natural_A X := \{ \neg_A \tau \mid \tau \notin X \}$$

$\neg(X \to \tau)$ is weakest inconsistent type

$$\neg_{A\to B}(X \to \tau) := (\natural_A X) \to (\neg_B \tau)$$

a) inconsisnt

Strongest consistent

Weakest inconsistent

$$\frac{X \parallel \natural X \qquad \tau \asymp \neg\tau}{(X \to \tau) \asymp (\natural X \to \neg\tau)}$$

b) weakest

Assume $(X \to \tau) \asymp (\bar{Y} \to \bar{\sigma})$

Then $X \parallel \bar{Y}$ and $\tau \asymp \bar{\sigma}$. So

Strongest consistent

$$\frac{\bar{Y} \succeq \natural X \qquad \bar{\sigma} \preceq \neg\tau}{(\bar{Y} \to \bar{\sigma}) \preceq (\natural X \to \neg\tau)}$$

Weakest inconsistent

Main Theorem

Theorem

- $\Gamma \nvdash M : \tau$ if and only if $\natural\Gamma \vdash M : \neg\tau$,
  where $\natural(x_1 : X_1, \ldots, x_n : X_n) := x_1 : (\natural X_1), \ldots, x_n : (\natural X_n)$
- Let $X = \{ \tau \mid \Gamma \vdash M : \tau \}$. Then

$$\natural\Gamma \vdash M : \natural X$$

Proof) By mutual induction on the structure of the term.

Main Theorem

For a closed term M,

$$\nVdash M : \tau \text{ iff } \Vdash M : \neg\tau$$

Theorem

- $\Gamma \nvdash M : \tau$ if and only if $\natural\Gamma \vdash M : \neg\tau$,
  where $\natural(x_1 : X_1, \ldots, x_n : X_n) := x_1 : (\natural X_1), \ldots, x_n : (\natural X_n)$

- Let $X = \{\, \tau \mid \Gamma \vdash M : \tau \,\}$. Then

$$\natural\Gamma \vdash M : \natural X$$

Proof) By mutual induction on the structure of the term.

Main Theorem

Theorem

- $\Gamma \nvdash M : \tau$ if and only if $\natural\Gamma \vdash M : \neg\tau$,
  where $\natural(x_1 : X_1, \ldots, x_n : X_n) := x_1 : (\natural X_1), \ldots, x_n : (\natural X_n)$
- Let $X = \{\, \tau \mid \Gamma \vdash M : \tau \,\}$. Then

$$\natural\Gamma \vdash M : \natural X$$

$$\Gamma \vdash M : X \quad \text{iff} \quad \natural\Gamma \vdash M : \natural X$$

under a certain condition

Pr m.

Remark

Only prime type judgements have negations

$$\Gamma \nvdash M : \tau \iff \natural\Gamma \vdash M : \neg\tau$$

Prime type

Negation of an intersection type judgement needs meta-level union

$$\Gamma \nvdash M : \bigwedge X \iff \exists \tau \in X.\ \natural\Gamma \vdash M : \neg\tau$$

Intersection type

Outline

- Reviewing a refinement intersection type system
- Negation of the type system
- Extensions
  - Additional constants (e.g. recursion)
  - Categorical formalisation
- Discussions

$\lambda^{\rightarrow}$ + constants

$$M, N ::= x \mid \lambda x^A . M \mid M\ M \mid c$$

It is easy to handle additional constants

provided that we have an affirmative type system

Affirmative side

The set of prime types for c

$$\frac{\tau \in T_c}{\Gamma \vdash c : \tau}$$

Negative side

$$\frac{\bar{\sigma} \in \natural T_c}{\bar{\Delta} \vdash c : \bar{\sigma}}$$

Example: recursion

Target language: $M, N ::= x \mid \lambda x^A.M \mid M\ M \mid \mathbf{Y}_A$

Affirmative side

$$\frac{\exists(Y \to \tau) \in X.\ \forall \sigma \in Y.\ \Gamma \vdash \mathbf{Y} : X \to \sigma}{\Gamma \vdash \mathbf{Y} : X \to \tau}\ \textbf{coinductive}$$

Negative side

$$\frac{\bar{\sigma} \in \natural\{\,\theta \mid\ \vdash \mathbf{Y} : \theta\,\}}{\bar{\Delta} \Vdash \mathbf{Y} : \bar{\sigma}}$$

Example: recursion

Target language: $M, N ::= x \mid \lambda x^A.M \mid M\ M \mid \mathbf{Y}_A$

Affirmative side

$$\frac{\exists (Y \to \tau) \in X.\ \forall \sigma \in Y.\ \Gamma \vdash \mathbf{Y} : X \to \sigma}{\Gamma \vdash \mathbf{Y} : X \to \tau}\ \textbf{coinductive}$$

Negative side

Equivalent to **the inductive version** of the above rule

$$\frac{\bar{\sigma} \in \natural\{\,\theta \mid\ \vdash \mathbf{Y} : \theta\,\}}{\bar{\Delta} \Vdash \mathbf{Y} : \bar{\sigma}}$$

Outline

- Reviewing a refinement intersection type system
- Negation of the type system
- Extensions
    - Additional constants (e.g. recursion)
    - Categorical formalisation
- Discussions

Semantics of terms via type system

Syntax	**Semantics**
Kind	Poset
A	$(\mathrm{Ty}_Q(A)/{\approx},\ \preceq)$
Term	Relation
$x :: A \vdash M :: B$	$\{\,(X, \tau) \mid x : X \vdash M : \tau\,\}$

Category $\mathbf{ScottL}_u$

Definition The category $\mathbf{ScottL}_u$ is given by:

Object: Poset $(A, \leq_A)$.

Morphism: An upward-closed relation

$$R \subseteq u(A)^{op} \times B$$

Composition: Let $\begin{matrix} R \subseteq u(A)^{op} \times B \\ S \subseteq u(B)^{op} \times C \end{matrix}$. Then

$$\frac{\exists Y \in u(B).\Big(\forall b \in Y.(X, b) \in R \text{ and } (Y, c) \in S\Big)}{(X, c) \in (S \circ R)}$$

Interpretation of CbN $\lambda^{\rightarrow}$ in $\mathbf{ScottL}_u$

Fact $\mathbf{ScottL}_u$ is a cartesian closed category.

Interpretation of kinds is given by:

$$[\![o]\!]_Q := (Q, =)$$
$$[\![A \rightarrow B]\!]_Q := u([\![A]\!]_Q)^{op} \times [\![B]\!]_Q$$

Hence $[\![A]\!]_Q \cong \mathrm{Ty}_Q(A)$.

Fact (see e.g. [Terui 2012])

$$\Gamma \vdash M : \tau \quad \Leftrightarrow \quad (\Gamma, \tau) \in [\![M]\!]$$

Negation Functor on $\mathbf{ScottL}_u$

The functor $\varphi \colon \mathbf{ScottL}_u \to \mathbf{ScottL}_u$ is defined by:

$$\varphi(A) := A^{op}$$
$$\varphi(R) := \{\, (A \backslash X, b) \in u(A)^{op} \times B \mid (X, b) \notin R \,\}$$

Lemma φ is an isomorphism on $\mathbf{ScottL}_u$.

If $\mathrm{R} \in u(A)^{op} \times B$ and $A = \emptyset$, then

$$\varphi(R) = \{\, (\emptyset, b) \mid (\emptyset, b) \notin R \,\}$$

which is essentially the complement of R.

Applications

$$\text{Negation} \quad \varphi : \mathbf{ScottL}_u \xrightarrow{\cong} \mathbf{ScottL}_u$$

- A type system witnessing call-by-value reachability [T&Kobayashi 14] is the Kleisli category of a monad

$$T : \mathbf{ScottL}_u \to \mathbf{ScottL}_u$$

Then

$$\varphi T \varphi^{-1} : \mathbf{ScottL}_u \to \mathbf{ScottL}_u$$

is also a monad. We can lift the negation to

$$\varphi : (\mathbf{ScottL}_u)_T \to (\mathbf{ScottL}_u)_{\varphi T \varphi^{-1}}$$

A type system proving unreachability

Applications

$$\text{Negation} \quad \varphi : \mathbf{ScottL}_u \xrightarrow{\cong} \mathbf{ScottL}_u$$

- A type system for higher-order model checking [Kobayashi&Ong 09] is coKleisli category of a comonad

$$\Box : \mathbf{ScottL}_u \to \mathbf{ScottL}_u$$ [Grellois&Melliès 14]

Then

$$\varphi \Box \varphi^{-1} : \mathbf{ScottL}_u \to \mathbf{ScottL}_u$$

is also a monad. We can lift the negation to

$$\varphi : (\mathbf{ScottL}_u)_{\Box} \to (\mathbf{ScottL}_u)_{\varphi \Box \varphi^{-1}}$$

Essentially the same as [Kobayashi&Ong 09]

Outline

- Reviewing a refinement intersection type system
- Negative type system
- Extensions
- Discussions

Automata complementation

Corresponds to negation of a 2nd-order judgement.

Boolean Closedness of Types

Let A be a kind and B_A be the set of all Böhm trees of type A. A language is a subset of B_A.

Definition A language $L \subseteq B_A$ is type-definable if there exists a type τ such that

$$L = \{ M \in B_A \mid \vdash M : \tau \}$$

in the type system for higher-order model checking [Kobayashi&Ong 09] [T&Ong 14].

Corollary The class of type-definable languages are closed under Boolean operations on sets.

Related Work

"Krivine machines and higher-order schemes"
[Salvati&Walkiewicz 12]

- The notion of consistency and inconsistency can be found in their work (called complementarity for the former and the latter has no name).
- This talk is partially inspired by their work.

Conclusion

Negation is a definable operation in the refinement intersection type system for the call-by-name $\lambda^{\rightarrow}$.

This observation leads to the construction of negative type systems for other refinement type systems, e.g.,

- call-by-name $\lambda^{\rightarrow}$ + recursion
- the type system for HOMC
- a type system for a call-by-value language

Application to verification needs some work.