Type-Checking & Type-Inference Problems for ∀ and ∃ Types

Koji Nakazawa
中澤 巧爾

Kyoto University
京都大学

joint work with
Makoto Tatsuta, Yuki Kato,
Yukiyoshi Kameyama, and Hiroshi Nakano

IRIT, Toulouse, Oct. 2009

This Work

- studies some decision problems on typability of programs with $\forall$- and $\exists$-types
  - Type Checking (TC): $\mathbf{\Gamma \vdash M : A?}$
    Does a given program have a given type?
  - Type Inference (TI): $\mathbf{? \vdash M :?}$
    Does a given program have any type?
- gives a new application of continuation-passing-style (CPS) translation
  - the problems for $\forall$ can be reduced to another problem for $\exists$

Contents

- Background
  - Type systems
  - $\forall$- and $\exists$-types
- TC and TI for $\forall$-types
- TC and TI for $\exists$-types
- Type-free-style $\boldsymbol{\lambda}$-calculi

Background: Type System

- A formal method to syntactically prove consistency of data types in programs

$$\mathbf{x_1 : A_1, x_2 : A_2, \cdots \vdash M : B}$$

= program **M** has type **B**

if variable $\mathbf{x_1}$ has type $\mathbf{A_1}$, and
variable $\mathbf{x_2}$ has type $\mathbf{A_2}$,...

- Examples
  - ✓ $\vdash \lambda x.x + 1$: int⟶int
  - ✓ $\vdash (\lambda x.x + 1)2$: int
  - × $\vdash (\lambda x.x + 1)$"abc" :??

Type soundness

If a program has a type, it is executed safely

Background: Type-Related Problems

Type Checking (TC) — $\Gamma \vdash M : A?$

For given Γ, M, and A, does $\Gamma \vdash M : A$ hold or not?
(Does a given program have a given type?)

Type Inference (TI) — $? \vdash M :?$

For given M, are there Γ and A such that $\Gamma \vdash M : A$ holds?
(Does a given program have any type?)

- TC and TI are decidable in
  - the simply typed λ-calculus
  - the type system of ML

Background: $\forall$- and $\exists$-Types

- Polymorphic types $\forall \mathbf{X.A}$
  - [Girard'72,Reynolds'74]
  - Types of polymorphic functions

    $\vdash \boldsymbol{\lambda x.x} : \forall \mathbf{X.X} \to \mathbf{X} \quad \Rightarrow \quad \vdash \mathbf{(\lambda x.x)2} : \mathsf{int}$

    $\vdash \boldsymbol{\lambda x.x} : \forall \mathbf{X.X} \to \mathbf{X} \quad \Rightarrow \quad \vdash \mathbf{(\lambda x.x)}\mathtt{true} : \mathsf{bool}$
- Existential types $\exists \mathbf{X.A}$

  $\vdash \mathbf{M : A[X := B]} \quad \Rightarrow \quad \vdash \langle \mathbf{B, M} \rangle \mathbf{: \exists X.A}$
  - Abstract data types [Mitchell&Plotkin'88]
    - ML module system
    - The type **B** used in the "implementation" **M** is hidden
  - Target of CPS translations of the polymorphic typed calculi [Hasegawa'05,Fujita'05]
    - $\neg \wedge \exists$-fragment $\boldsymbol{\lambda}^{\neg\wedge\exists}$

Results

TC and TI in domain-free λ-calculi

$$
\begin{array}{ccccc}
 & & \text{TC in DF-}\boldsymbol{\lambda}^{\neg\wedge\exists} & \simeq^{[3]} & \text{TI in DF-}\boldsymbol{\lambda}^{\neg\wedge\exists} \\
 & & \vee\!|^{[1]} & & \vee\!|^{[1]} \\
\text{2UP} & \leq^{[4]} & \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} & \simeq^{[2]} & \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} \\
 & & |\!\wedge^{[1]} & & |\!\wedge^{[1]} \\
 & & \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\exists} & \simeq^{[3]} & \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\exists}
\end{array}
$$

[1] Nakazawa&Tatsuta&Kameyama&Nakano'08
[2] Nakazawa&Tatsuta'09.
[3] Kato&Nakazawa'09.
[4] Fujita&Schubert'00.

- $\mathcal{P} \leq \mathcal{Q}$ iff $\mathcal{P}$ is Turing reducible to $\mathcal{Q}$
  - decidability of $\mathcal{Q}$ implies decidability of $\mathcal{P}$
  - undecidability of $\mathcal{P}$ implies undecidability of $\mathcal{Q}$
- $\mathcal{P} \simeq \mathcal{Q}$ iff $\mathcal{P} \leq \mathcal{Q}$ and $\mathcal{Q} \leq \mathcal{P}$

Contents

- Background
- TC and TI for $\forall$-types
  - Curry style
  - Domain-free style
- TC and TI for $\exists$-types
- Type-free-style $\boldsymbol{\lambda}$-calculi

Polymorphic Lambda Calculus

$\lambda^{\to\forall}$ Curry style

Types: $\mathbf{A ::= X \mid A \to B \mid \forall X.A}$

$$\frac{}{\Gamma, x : A \vdash x : A}\ (\mathrm{Ax})$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}\ (\to\mathrm{I}) \qquad \frac{\Gamma \vdash M : A \to B \quad \Delta \vdash N : A}{\Gamma, \Delta \vdash MN : B}\ (\to\mathrm{E})$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash M : \forall X.A}\ (\forall\mathrm{I})^* \qquad \frac{\Gamma \vdash M : \forall X.A}{\Gamma \vdash M : A[X := B]}\ (\forall\mathrm{E})$$

*$\mathbf{X}$ must not be contained in Γ

- $\vdash \lambda x.x : \forall X.X \to X$

In Curry-Style $\boldsymbol{\lambda}^{\rightarrow\forall}$

Theorem [Wells'94]

TC and TI in Curry-$\boldsymbol{\lambda}^{\rightarrow\forall}$ are equivalent and undecidable

$$\text{SUP} \quad \leq \quad \text{TC in Curry-}\boldsymbol{\lambda}^{\rightarrow\forall} \quad \simeq \quad \text{TI in Curry-}\boldsymbol{\lambda}^{\rightarrow\forall}$$

- SUP = semi-unification problem
  - undecidability is proved in [Kfoury et al.'93]

From Curry to Domain-Free

- How about other formulation where programs contain more type information?

$\lambda^{\to\forall}$ Curry style

Types: $\mathbf{A ::= X \mid A \to B \mid \forall X.A}$

$$\frac{}{\Gamma, x : A \vdash x : A}\ (\mathrm{Ax})$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}\ (\to\mathrm{I}) \qquad \frac{\Gamma \vdash M : A \to B \quad \Delta \vdash N : A}{\Gamma, \Delta \vdash MN : B}\ (\to\mathrm{E})$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash M : \forall X.A}\ (\forall\mathrm{I})^* \qquad \frac{\Gamma \vdash M : \forall X.A}{\Gamma \vdash M : A[X := B]}\ (\forall\mathrm{E})$$

*$\mathbf{X}$ must not be contained in Γ

From Curry to Domain-Free

- How about other formulation where programs contain more type information?

$\lambda^{\to\forall}$ domain-free style

Types: $\mathbf{A ::= X \mid A \to B \mid \forall X.A}$

$$\frac{}{\Gamma, x : A \vdash x : A}\ (\mathrm{Ax})$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}\ (\to\mathrm{I}) \qquad \frac{\Gamma \vdash M : A \to B \quad \Delta \vdash N : A}{\Gamma, \Delta \vdash MN : B}\ (\to\mathrm{E})$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \Lambda X.M : \forall X.A}\ (\forall\mathrm{I})^* \qquad \frac{\Gamma \vdash M : \forall X.A}{\Gamma \vdash MB : A[X := B]}\ (\forall\mathrm{E})$$

*$\mathbf{X}$ must not be contained in Γ

In Domain-Free $\boldsymbol{\lambda}^{\rightarrow\forall}$

Theorem [Fujita&Schubert'00]

TC in DF-$\boldsymbol{\lambda}^{\rightarrow\forall}$ is undecidable

$$\text{2UP} \quad \leq \quad \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\forall}$$

- 2UP = 2nd-order-unification problem
  - undecidability is proved in [Schubert'98]

Theorem [Nakazawa&Tatsuta'09]

TC and TI in DF-$\boldsymbol{\lambda}^{\rightarrow\forall}$ are equivalent and undecidable

$$\text{2UP} \quad \leq \quad \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} \quad \simeq \quad \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\forall}$$

TC $\leq$ TI in DF-$\lambda^{\rightarrow\forall}$

Proposition [Nakazawa&Tatsuta'09]

For any program $\mathbf{M}$ and any type $\mathbf{A}$,
we can effectively construct a program $\mathbf{J_{M,A}}$ such that

$$\vdash \mathbf{M : A} \quad \text{iff} \quad \vdash \mathbf{J_{M,A} : B} \text{ for some } \mathbf{B}$$

Proof $\mathbf{J_{M,A} = \lambda x.(\lambda y.x(A \to \bot)M)(x(\bot \to \bot)x)}$

(where $\bot \equiv \forall \mathbf{X.X}$)

$$\dfrac{\dfrac{\dfrac{}{\mathbf{x : \bot \vdash x : \bot}}}{\mathbf{x : \bot \vdash x(A \to \bot) : A \to \bot}} \quad \vdots \; \mathbf{\vdash M : A}}{\mathbf{x : \bot \vdash x(A \to \bot)M : \bot}}$$

- Key fact:

  if $\mathbf{x(\bot \to \bot)x}$ is typable, the type of $\mathbf{x}$ must be $\bot$

Roadmap

TC and TI in domain-free λ-calculi

$$
\begin{array}{ccccc}
 & & \text{TC in DF-}\boldsymbol{\lambda}^{\neg\wedge\exists} & \simeq & \text{TI in DF-}\boldsymbol{\lambda}^{\neg\wedge\exists} \\
 & & \vee| & & \vee| \\
\text{2UP} & \leq & \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} & \simeq & \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} \\
 & & |\wedge & & |\wedge \\
 & & \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\exists} & \simeq & \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\exists}
\end{array}
$$

Contents

- Background
- TC and TI for $\forall$-types
- TC and TI for $\exists$-types
  - $\neg \wedge \exists$-fragment
  - From $\forall$ to $\exists$
- Type-free-style $\boldsymbol{\lambda}$-calculi

$\neg \wedge \exists$-Fragment

- Tatsuta has proved

Theorem [Tatsuta et al.'08]

The inhabitation in $\boldsymbol{\lambda}^{\neg\wedge\exists}$ is decidable

Inhabitation problem ($=$ provability problem):

Is there some program which has a given type?
($\vdash \mathbf{?} : \mathbf{A}$)

- How about TC and TI in $\boldsymbol{\lambda}^{\neg\wedge\exists}$?
  - TC and TI in DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}$ are undecidable

Domain-Free $\lambda^{\neg\wedge\exists}$

$\lambda^{\neg\wedge\exists}$ domain-free style

Types: $\mathbf{A ::= X \mid \bot \mid \neg A \mid A \wedge B \mid \exists X.A}$

$$\frac{}{\Gamma, x : A \vdash x : A}\ (\mathrm{Ax})$$

$$\frac{\Gamma, x : A \vdash M : \bot}{\Gamma \vdash \lambda x.M : \neg A}\ (\neg\mathrm{I}) \qquad \frac{\Gamma \vdash M : \neg A \quad \Delta \vdash N : A}{\Gamma, \Delta \vdash MN : \bot}\ (\neg\mathrm{E})$$

$$\frac{\Gamma \vdash M : A \quad \Delta \vdash N : B}{\Gamma, \Delta \vdash \langle M, N\rangle : A \wedge B}\ (\wedge\mathrm{I}) \qquad \frac{\Gamma \vdash M : A_1 \wedge A_2}{\Gamma \vdash M\pi_i : A_i}\ (\wedge\mathrm{E})$$

$$\frac{\Gamma \vdash N : A[X := B]}{\Gamma \vdash \langle B, N\rangle : \exists X.A}\ (\exists\mathrm{I})$$

$$\frac{\Gamma \vdash M : \exists X.A \quad \Delta, x : A \vdash N : C}{\Gamma, \Delta \vdash M[Xx.N] : C}\ (\exists\mathrm{E})^*$$

*$\mathbf{X}$ must not be contained in $\mathbf{\Delta}$ and $\mathbf{C}$

TC and TI for ∃-types

Theorems

1. [Nakazawa et al.'08]
   TC and TI are undecidable in DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}/\boldsymbol{\lambda}^{\rightarrow\exists}$
2. [Kato&Nakazawa'09]
   TC and TI are equivalent in DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}/\boldsymbol{\lambda}^{\rightarrow\exists}$

$$
\begin{array}{ccccc}
 & & \text{TC in DF-}\boldsymbol{\lambda}^{\neg\wedge\exists} & \simeq^{\mathbf{2}} & \text{TI in DF-}\boldsymbol{\lambda}^{\neg\wedge\exists} \\
 & & \vee|^{\mathbf{1}} & & \vee|^{\mathbf{1}} \\
\text{2UP} & \leq & \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} & \simeq & \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} \\
 & & |\wedge^{\mathbf{1}} & & |\wedge^{\mathbf{1}} \\
 & & \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\exists} & \simeq^{\mathbf{2}} & \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\exists}
\end{array}
$$

From $\forall$ to $\exists$

Goal

TC in DF-$\boldsymbol{\lambda}^{\rightarrow\forall} \leq$ TC in DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}$

- It is sufficient to find a computable translation

  $\mathbf{F}$: DF-$\boldsymbol{\lambda}^{\rightarrow\forall} \longrightarrow$ DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}$

  $$(\rightarrow, \forall)\text{-types} \longrightarrow (\neg, \wedge, \exists)\text{-types}$$
  $$\boldsymbol{\lambda}^{\rightarrow\forall}\text{-programs} \longrightarrow \boldsymbol{\lambda}^{\neg\wedge\exists}\text{-programs}$$

  such that

  $$\mathbf{\Gamma} \vdash_{\boldsymbol{\lambda}^{\rightarrow\forall}} \mathbf{M : A} \quad \text{iff} \quad \mathbf{F(\Gamma)} \vdash_{\boldsymbol{\lambda}^{\neg\wedge\exists}} \mathbf{F(M) : F(A)}$$

CPS translation

- Continuation-passing-style (CPS) translation
  - is a program translation
  - makes an evaluation order explicit
  - preserves "meaning" of programs
    - equality
    - reduction
    - typability
- $\overline{\mathbf{M}}, \overline{\mathbf{A}}$: DF-$\boldsymbol{\lambda}^{\rightarrow\forall} \longrightarrow$ DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}$

$$(\rightarrow, \forall)\text{-types} \longrightarrow (\neg, \wedge, \exists)\text{-types}$$

$$\text{DF-}\boldsymbol{\lambda}^{\rightarrow\forall}\text{-programs} \longrightarrow \text{DF-}\boldsymbol{\lambda}^{\neg\wedge\exists}\text{-programs}$$

which preserves typability of programs

CPS translation

CPS translation [Fujita'05,Hasegawa'05]

$\overline{\mathbf{A}}$: $(\to, \forall)$-types $\longrightarrow$ CPS types $(\subset (\neg, \wedge, \exists)$-types)
$\overline{\mathbf{M}}$: DF-$\lambda^{\to\forall}$-programs $\longrightarrow$ CPS programs $(\subset$ DF-$\lambda^{\neg\wedge\exists})$

$$\overline{\mathbf{A}} \equiv \neg\mathbf{A}^* \qquad \overline{\mathbf{x}} \equiv \lambda\mathbf{k}.\mathbf{x}\mathbf{k}$$

$$\overline{\lambda\mathbf{x}.\mathbf{M}} \equiv \lambda\mathbf{k}.(\lambda\mathbf{x}.\overline{\mathbf{M}}(\mathbf{k}\pi_2))(\mathbf{k}\pi_1)$$

$$\mathbf{X}^* \equiv \mathbf{X} \qquad \overline{\mathbf{M}\mathbf{N}} \equiv \lambda\mathbf{k}.\overline{\mathbf{M}}\langle\overline{\mathbf{N}}, \mathbf{k}\rangle$$

$$(\mathbf{A} \to \mathbf{B})^* \equiv \neg\overline{\mathbf{A}} \wedge \overline{\mathbf{B}} \qquad \overline{\Lambda\mathbf{X}.\mathbf{M}} \equiv \lambda\mathbf{k}.\mathbf{k}[\mathbf{X}\mathbf{k}'.\overline{\mathbf{M}}\mathbf{k}']$$

$$(\forall\mathbf{X}.\mathbf{A})^* \equiv \exists\mathbf{X}.\overline{\mathbf{A}} \qquad \overline{\mathbf{M}\mathbf{A}} \equiv \lambda\mathbf{k}.\overline{\mathbf{M}}\langle\overline{\mathbf{A}}, \mathbf{k}\rangle$$

Preservation of typability

$$\Gamma \vdash_{\lambda^{\to\forall}} \mathbf{M} : \mathbf{A} \quad \Rightarrow \quad \overline{\Gamma} \vdash_{\lambda^{\neg\wedge\exists}} \overline{\mathbf{M}} : \overline{\mathbf{A}}$$

From TC for $\forall$ to TC for $\exists$

- Our goal is

$$\Gamma \vdash_{\lambda^{\to\forall}} M : A \quad \text{iff} \quad \overline{\Gamma} \vdash_{\lambda^{\neg\wedge\exists}} \overline{M} : \overline{A}$$

Preservation of typability

$$\Gamma \vdash_{\lambda^{\to\forall}} M : A \quad \text{iff} \quad \overline{\Gamma} \vdash_{\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}} \overline{M} : \overline{A}$$

- $\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}$ is the image of the CPS-translation consisting of
  - CPS types $\mathcal{A}$ and CPS terms P
  - type judgments $x : \mathcal{A}, \cdots \vdash P : \mathcal{B}$

$\lambda^{\neg\wedge\exists}$ is conservative over $\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}$

$$\overline{\Gamma} \vdash_{\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}} \overline{M} : \overline{A} \quad \text{iff} \quad \overline{\Gamma} \vdash_{\lambda^{\neg\wedge\exists}} \overline{M} : \overline{A}$$

Conservativeness

$\lambda^{\neg\wedge\exists}$ is conservative over $\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}$

$$\overline{\Gamma} \vdash_{\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}} \overline{M} : \overline{A} \quad \Leftarrow \quad \overline{\Gamma} \vdash_{\lambda^{\neg\wedge\exists}} \overline{M} : \overline{A}$$

- In the following derivation in $\lambda^{\neg\wedge\exists}$

$$\dfrac{\begin{array}{c}\vdots\\ \vdash \overline{M} : \neg(A \wedge \overline{B})\end{array} \quad \dfrac{\begin{array}{c}\vdots\\ \vdash \overline{N} : A\end{array} \quad \overline{k : \overline{B} \vdash k : \overline{B}}}{k : \overline{B} \vdash \langle \overline{N}, k \rangle : A \wedge \overline{B}}}{\dfrac{k : \overline{B} \vdash \overline{M}\langle \overline{N}, k \rangle : \bot}{\vdash \lambda k.\overline{M}\langle \overline{N}, k \rangle : \overline{B}}}$$

  A may not be a CPS type, and then the derivation is not in $\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}$

Type Contraction

Type contraction

$(\neg \mathbf{A})^{\mathbf{c}} \equiv \neg \mathbf{A}^{\mathbf{d}}$

$\mathbf{A}^{\mathbf{c}} \equiv \neg \mathbf{A}^{\mathbf{d}}$ ($\mathbf{A}$ is not a negation)

$\mathbf{X}^{\mathbf{d}} \equiv \mathbf{X}$

$\bot^{\mathbf{d}} \equiv \exists \mathbf{X}.\mathbf{X}$

$(\neg \mathbf{A})^{\mathbf{d}} \equiv \mathbf{A}^{\mathbf{d}}$

$(\mathbf{A} \wedge \mathbf{B})^{\mathbf{d}} \equiv \mathbf{A}^{\mathbf{c}} \wedge \mathbf{B}^{\mathbf{d}}$

$(\exists \mathbf{X}.\mathbf{A})^{\mathbf{d}} \equiv \exists \mathbf{X}.\mathbf{A}^{\mathbf{d}}$

Key Lemma

1. $\mathbf{A}^{\mathbf{c}}$ is a CPS type
2. $\mathcal{A}^{\mathbf{c}} = \mathcal{A}$
3. $\mathbf{\Gamma} \vdash_{\lambda^{\neg\wedge\exists}} \overline{\mathbf{M}} : \mathbf{A} \Rightarrow \mathbf{\Gamma}^{\mathbf{c}} \vdash_{\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}} \overline{\mathbf{M}} : \mathbf{A}^{\mathbf{c}}$

Proposition

$\overline{\mathbf{\Gamma}} \vdash_{\lambda^{\neg\wedge\exists}} \overline{\mathbf{M}} : \overline{\mathbf{A}} \quad \Rightarrow \quad \overline{\mathbf{\Gamma}} \vdash_{\lambda^{\neg\wedge\exists}_{\mathbf{CPS}}} \overline{\mathbf{M}} : \overline{\mathbf{A}}$

From $\forall$ to $\exists$

Theorem [Nakazawa et al.'08]

1. TC in DF-$\boldsymbol{\lambda}^{\rightarrow\forall} \leq$ TC in DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}$
2. TI in DF-$\boldsymbol{\lambda}^{\rightarrow\forall} \leq$ TI in DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}$
3. TC in DF-$\boldsymbol{\lambda}^{\rightarrow\forall} \leq$ TC in DF-$\boldsymbol{\lambda}^{\rightarrow\exists}$
4. TI in DF-$\boldsymbol{\lambda}^{\rightarrow\forall} \leq$ TI in DF-$\boldsymbol{\lambda}^{\rightarrow\exists}$

- We can use CPS translations from DF-$\boldsymbol{\lambda}^{\rightarrow\forall}$ to DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}$/DF-$\boldsymbol{\lambda}^{\rightarrow\exists}$

Summary for DF calculi

TC and TI in domain-free $\boldsymbol{\lambda}$-calculi

$$
\begin{array}{ccccc}
 & & \text{TC in DF-}\boldsymbol{\lambda}^{\neg\wedge\exists} & \simeq & \text{TI in DF-}\boldsymbol{\lambda}^{\neg\wedge\exists} \\
 & & \vee\mathrm{I} & & \vee\mathrm{I} \\
\text{2UP} & \leq & \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} & \simeq & \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\forall} \\
 & & \wedge\mathrm{I} & & \wedge\mathrm{I} \\
 & & \text{TC in DF-}\boldsymbol{\lambda}^{\rightarrow\exists} & \simeq & \text{TI in DF-}\boldsymbol{\lambda}^{\rightarrow\exists}
\end{array}
$$

Theorem [Kato&Nakazawa '09]

1. TC $\simeq$ TI in DF-$\boldsymbol{\lambda}^{\neg\wedge\exists}$
2. TC $\simeq$ TI in DF-$\boldsymbol{\lambda}^{\rightarrow\exists}$

Contents

- Background
- TC and TI for $\forall$-types
- TC and TI for $\exists$-types
- Type-free-style $\boldsymbol{\lambda}$-calculi

Type-Free-Style $\lambda^{\exists}$

TF-$\lambda^{\exists}$ [Tatsuta'07,Fujita&Schubert'09]

$$\frac{\Gamma \vdash N : A[X := B]}{\Gamma \vdash \langle \exists, N \rangle : \exists X.A}\ (\exists \mathrm{I}) \qquad \frac{\Gamma \vdash M : \exists X.A \quad \Delta, x : A \vdash N : C}{\Gamma, \Delta \vdash M[x.N] : C}\ (\exists \mathrm{E})$$

DF-$\lambda^{\exists}$

$$\frac{\Gamma \vdash N : A[X := B]}{\Gamma \vdash \langle B, N \rangle : \exists X.A}\ (\exists \mathrm{I}) \qquad \frac{\Gamma \vdash M : \exists X.A \quad \Delta, x : A \vdash N : C}{\Gamma, \Delta \vdash M[Xx.N] : C}\ (\exists \mathrm{E})$$

Summary for TF calculi

TC and TI in type-free-style $\boldsymbol{\lambda}$-calculi

$$
\begin{array}{ccccccc}
 & & \text{TC in TF-}\boldsymbol{\lambda}^{\neg\wedge\exists} & \geq^{[2]} & \text{TI in TF-}\boldsymbol{\lambda}^{\neg\wedge\exists} & & \\
 & & \vee|^{[1]} & & \vee|^{[1]} & & \\
 & & \text{TC in TF-}\boldsymbol{\lambda}^{\rightarrow\forall} & \geq & \text{TI in TF-}\boldsymbol{\lambda}^{\rightarrow\forall} & & \\
\text{2UP} & \leq^{[3]} & \text{TC in TF-}\boldsymbol{\lambda}^{\rightarrow\exists} & \geq & \text{TI in TF-}\boldsymbol{\lambda}^{\rightarrow\exists} & \geq^{[3]} & \text{2UP}
\end{array}
$$

[1] Nakazawa&Tatsuta'09 (by CPS translation)
[2] We can use the technique of Kato&Nakazawa'09
[3] Fujita&Schubert'09

Conclusion

- TC and TI for $\forall$- and $\exists$-types have been studied
  - undecidability in the domain-free-style $\boldsymbol{\lambda}^{\rightarrow\forall}$, $\boldsymbol{\lambda}^{\rightarrow\exists}$, and $\boldsymbol{\lambda}^{\neg\wedge\exists}$
  - some results for the type-free-style calculi
- CPS translation can be used for the decidability issue for the type-related problems

Future Work

- TC and TI in TF-$\boldsymbol{\lambda}^{\rightarrow\forall}/\boldsymbol{\lambda}^{\neg\wedge\exists}$?

TC and TI in type-free-style $\boldsymbol{\lambda}$-calculi

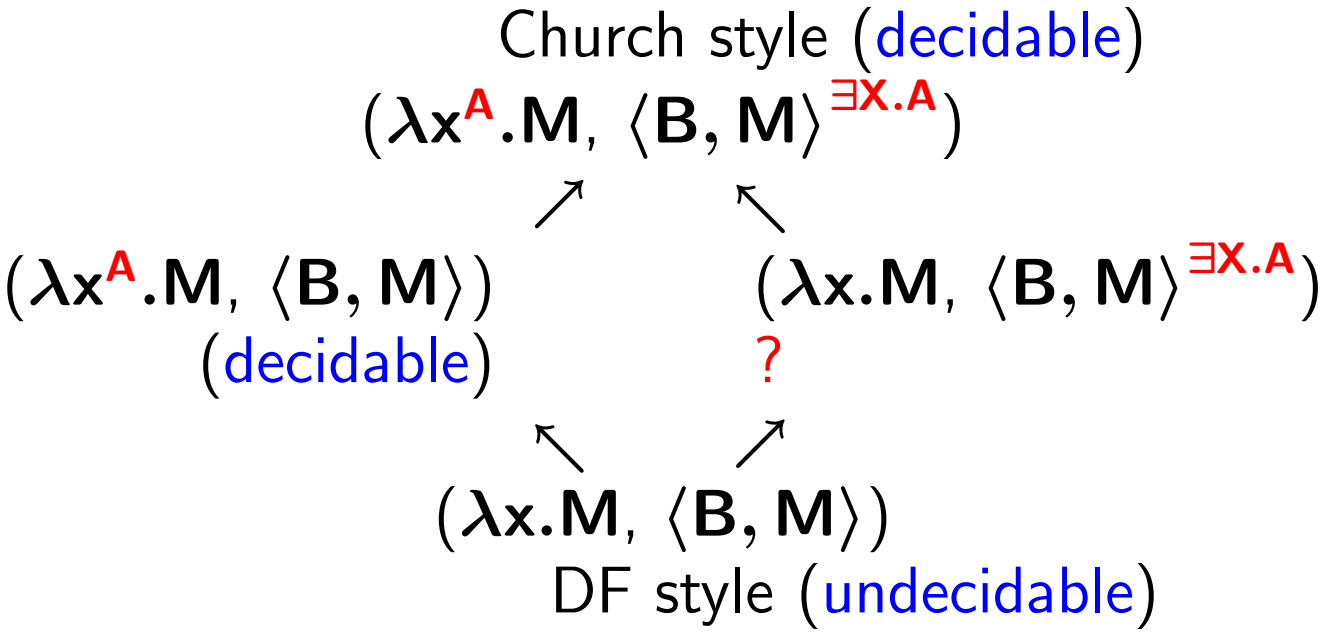
$$
\begin{array}{ccccccc}
 & & \text{TC in TF-}\boldsymbol{\lambda}^{\neg\wedge\exists} & \geq & \text{TI in TF-}\boldsymbol{\lambda}^{\neg\wedge\exists} & & \\
 & & \vee\vert & & \vee\vert & & \\
\text{2UP} & \leq^{?} & \text{TC in TF-}\boldsymbol{\lambda}^{\rightarrow\forall} & \geq & \text{TI in TF-}\boldsymbol{\lambda}^{\rightarrow\forall} & \geq^{?} & \text{2UP} \\
 & & & & & & \\
\text{2UP} & \leq & \text{TC in TF-}\boldsymbol{\lambda}^{\rightarrow\exists} & \geq & \text{TI in TF-}\boldsymbol{\lambda}^{\rightarrow\exists} & \geq & \text{2UP}
\end{array}
$$

Future Work

- How about other styles for $\exists$, where programs contain more type information?

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^{A}.M : A \to B}\ (\to \mathrm{I}) \qquad \frac{\Gamma \vdash M : A[X := B]}{\Gamma \vdash \langle B, M \rangle^{\exists X.A} : \exists X.A}\ (\exists \mathrm{I})$$

$\langle B, M \rangle^{\exists X.A}$ can be considered as signatures of ML modules



Thanks for your attention

ご静聴ありがとうございました