Complete and easy bidirectional typechecking for higher-rank polymorphism

Joshua Dunfield   Neelakantan R. Krishnaswami*

Max Planck Institute for Software Systems

Max Planck Institute for Software Systems
Kaiserslautern and Saarbrücken, Germany

[* ⟶ University of Birmingham]

ICFP 2013

Damas-Milner type inference

- The standard algorithm for inferring types using first-order unification, with **prenex polymorphism**:
  - $\forall$ on outside
  - predicative: instantiate $\forall$ with $\forall$-free types
- Excellent fit between algorithm and type system

Damas-Milner type inference

- The standard algorithm for inferring types using first-order unification, with **prenex polymorphism**:
  - $\forall$ on outside
  - predicative: instantiate $\forall$ with $\forall$-free types
- Excellent fit between algorithm and type system
- DM gives **confusing error messages**

Damas-Milner type inference

- The standard algorithm for inferring types using first-order unification, with **prenex polymorphism**:
  - $\forall$ on outside
  - predicative: instantiate $\forall$ with $\forall$-free types
- Excellent fit between algorithm and type system
- DM gives **confusing error messages**
- DM is based on full inference, often undecidable; **tricky to extend** to higher-rank polymorphism, modules, subtyping, . . . , which need some kind of type **checking**

Bidirectional typechecking

- Checking and synthesis:

$$\begin{array}{ll} \Gamma \vdash e \Leftarrow A & e \text{ checks against type } A \\ \Gamma \vdash e \Rightarrow A & e \text{ synthesizes type } A \end{array}$$

- Unification not a fundamental mechanism (sometimes not used at all!)

Bidirectional typechecking

- 1990s: Pierce & Turner ("local type inference")
- Can support many type system features:
  - refinements, indexed types, intersections and unions (Davies, Dunfield, Xi, Pfenning)
  - contextual modal types (Nanevski, Pientka, ...)
  - object-oriented subtyping, e.g. C$^\sharp$, Scala
- Can be remarkably clean, but also *ad hoc*
  - how many type annotations? where?

Bidirectional typechecking, proof-theoretically

A **proof-theoretic** recipe for bidirectional type systems:
Davies & Pfenning (2000); Dunfield & Pfenning (2004)

Introduction rules: checking $\Leftarrow$

$$\frac{\Gamma, x : A \vdash e \Leftarrow B}{\Gamma \vdash \lambda x.\, e \Leftarrow A \to B}$$

Elimination rules: synthesis $\Rightarrow$

$$\frac{\Gamma \vdash e_1 \Rightarrow A \to B \qquad \Gamma \vdash e_2 \Leftarrow A}{\Gamma \vdash e_1\, e_2 \Rightarrow B}$$

Bidirectional typechecking, proof-theoretically

A **proof-theoretic** recipe for bidirectional type systems:
Davies & Pfenning (2000); Dunfield & Pfenning (2004)

Introduction rules: checking $\Leftarrow$

$$\frac{\Gamma, x : A \vdash e \Leftarrow B}{\Gamma \vdash \lambda x.\, e \Leftarrow A \to B}$$

Elimination rules: synthesis $\Rightarrow$

$$\frac{\Gamma \vdash e_1 \Rightarrow A \to B \qquad \Gamma \vdash e_2 \Leftarrow A}{\Gamma \vdash e_1\, e_2 \Rightarrow B}$$

Consequence: **annotations only on redexes** $(\lambda x.\, e) e_2$
mediated by annotation rule

Bidirectional typechecking, proof-theoretically

A **proof-theoretic** recipe for bidirectional type systems:
Davies & Pfenning (2000); Dunfield & Pfenning (2004)

Introduction rules: checking $\Leftarrow$

$$\frac{\Gamma, x : A \vdash e \Leftarrow B}{\Gamma \vdash \lambda x.\, e \Leftarrow A \rightarrow B}$$

Elimination rules: synthesis $\Rightarrow$

$$\frac{\Gamma \vdash e_1 \Rightarrow A \rightarrow B \qquad \Gamma \vdash e_2 \Leftarrow A}{\Gamma \vdash e_1\, e_2 \Rightarrow B}$$

Annotation rule

$$\frac{\Gamma \vdash e \Leftarrow A}{\Gamma \vdash (e : A) \Rightarrow A}$$

Consequence: **annotations only on redexes** $(\lambda x.\, e)e_2$ mediated by annotation rule

Something is missing

Bidirectional typechecking...

- Can support many type system features

Something is missing

Bidirectional typechecking. . .

- Can support many type system features

What about polymorphism?

$$\frac{\Gamma, \alpha \vdash e \Leftarrow A}{\Gamma \vdash e \Leftarrow \forall\alpha.\, A}\ \forall\mathrm{I} \qquad \frac{\Gamma \vdash e \Rightarrow \forall\alpha.\, A \qquad \Gamma \vdash \tau}{\Gamma \vdash e \Rightarrow [\tau/\alpha]A}\ \forall\mathrm{E}$$

Can't magically guess τ.

Something is missing

Bidirectional typechecking…

- Can support many type system features

What about polymorphism?

$$\frac{\Gamma, \alpha \vdash e \Leftarrow A}{\Gamma \vdash e \Leftarrow \forall \alpha.\, A}\ \forall\mathrm{I} \qquad \frac{\Gamma \vdash e \Rightarrow \forall \alpha.\, A \qquad \Gamma \vdash \tau}{\Gamma \vdash e \Rightarrow [\tau/\alpha]A}\ \forall\mathrm{E}$$

Can't magically guess τ.

- Davies' thesis: also do Damas-Milner inference

How can we cleanly express the
instantiation of polymorphism ($\forall$E)
in a bidirectional system?

Something is missing

Bidirectional typechecking...

- Can support many type system features

What about polymorphism?

$$\frac{\Gamma, \alpha \vdash e \Leftarrow A}{\Gamma \vdash e \Leftarrow \forall \alpha.\, A}\ \forall I \qquad \frac{\Gamma \vdash e \Rightarrow \forall \alpha.\, A \qquad \Gamma \vdash \tau}{\Gamma \vdash e \Rightarrow [\tau/\alpha]A}\ \forall E$$

Can't magically guess τ.

- Davies' thesis: also do Damas-Milner inference

How can we cleanly express the
instantiation of polymorphism ($\forall$E)
in a bidirectional system?

- **This paper:**
  a "complete" and "easy" answer to this question

Contributions

- A bidirectional version of predicative System F, capturing the **bidirectionality** of our algorithm but not our method of instantiating $\forall$ quantifiers

  (the **declarative system**)

  - the system that programmers need to understand
  - annotations only on **redexes of polymorphic type**

Contributions

- A bidirectional version of predicative System F, capturing the **bidirectionality** of our algorithm but not our method of instantiating ∀ quantifiers

  (the **declarative system**)

  - the system that programmers need to understand
  - annotations only on **redexes of polymorphic type**

- A bidirectional typing algorithm that expresses how to instantiate ∀

  (the **algorithmic system**)

  - **Sound** and **complete** with respect to the declarative system: **users need not understand the full algorithm**

Declarative system: Syntax

Terms	e	$::=$	$x \mid () \mid \lambda x.\, e \mid e\; e \mid (e : A)$
Types	A, B, C	$::=$	$1 \mid \alpha \mid \forall \alpha.\, A \mid A \to B$
Monotypes	τ, σ	$::=$	$1 \mid \alpha \mid \sigma \to \tau$
Contexts	Ψ	$::=$	$\cdot \mid \Psi, \alpha \mid \Psi, x : A$

- Distinguish monotypes τ, σ to enforce predicativity, but allow higher-rank polymorphism

Declarative system: Typing judgments

Checking

$\Psi \vdash e \Leftarrow A$ e checks against type A

Synthesis

$\Psi \vdash e \Rightarrow A$ e synthesizes type A

Application (à la spine form)

$\Psi \vdash A \bullet e \Rrightarrow C$ applying term of type A
to arg. e synthesizes C

Declarative system: Typing judgments

Checking

$\Psi \vdash e \Leftarrow A$ e checks against type A

Synthesis

$\Psi \vdash e \Rightarrow A$ e synthesizes type A

Application (à la spine form)

$\Psi \vdash A \bullet e \Rrightarrow C$ applying term of type A to arg. e synthesizes C

$$\frac{\Psi \vdash e_1 \Rightarrow A \qquad \Psi \vdash A \bullet e_2 \Rrightarrow C}{\Psi \vdash e_1\, e_2 \Rightarrow C}\ \mathsf{Decl}{\to}\mathsf{E}$$

Lets us instantiate $\forall$ only where necessary

Declarative system: Some typing rules

$$\frac{(x : A) \in \Psi}{\Psi \vdash x \Rightarrow A}\ \mathsf{DeclVar} \qquad \frac{\Psi \vdash e \Rightarrow A \qquad \Psi \vdash A \leq B}{\Psi \vdash e \Leftarrow B}\ \mathsf{DeclSub}$$

$$\frac{\Psi, x : A \vdash e \Leftarrow B}{\Psi \vdash \lambda x.e \Leftarrow A \rightarrow B}\ \mathsf{Decl{\rightarrow}I} \qquad \frac{\Psi \vdash e_1 \Rightarrow A \qquad \Psi \vdash A \bullet e_2 \Rrightarrow C}{\Psi \vdash e_1\, e_2 \Rightarrow C}\ \mathsf{Decl{\rightarrow}E}$$

$$\frac{\Psi \vdash \sigma \rightarrow \tau \qquad \Psi, x : \sigma \vdash e \Leftarrow \tau}{\Psi \vdash \lambda x.\, e \Rightarrow \sigma \rightarrow \tau}\ \mathsf{Decl{\rightarrow}I{\Rightarrow}}$$

Declarative system: Subtyping

- Include (weak!) subtyping à la Odersky–Läufer:

  "$A \leq B$ if A is at least as polymorphic as B"

  or

  "$A \leq B$ if B instantiates (some) quantifiers in A"

  (respecting contravariance)

Declarative system: Subtyping

- Include (weak!) subtyping à la Odersky–Läufer:

  "$A \leq B$ if A is at least as polymorphic as B"

  or

  "$A \leq B$ if B instantiates (some) quantifiers in A"

  (respecting contravariance)

$$\frac{\Psi \vdash \tau \qquad \Psi \vdash [\tau/\alpha]A \leq B}{\Psi \vdash \forall \alpha.\, A \leq B} \leq\!\forall\mathsf{L} \qquad \frac{\Psi, \beta \vdash A \leq B}{\Psi \vdash A \leq \forall \beta.\, B} \leq\!\forall\mathsf{R}$$

Declarative system: metatheory

- Define predicative System F with implicit poly.:

$$\Psi \vdash e : A$$

  (6 rules; see paper)

- Declarative system *complete* (annotatability)
- Declarative system *sound* (η-expansion)

Algorithmic system

- Directly* implementable rules, but
  sound and **complete**
  with respect to our declarative system
- Judgments will include **input** and **output** contexts

* Actually true.

Algorithmic system: Syntax

Introduces **existential** type variables $\hat{\alpha}$

Types	A, B, C	$::=$	$1 \mid \alpha \mid \hat{\alpha} \mid \forall \alpha.\, A \mid A \to B$
Monotypes	τ, σ	$::=$	$1 \mid \alpha \mid \hat{\alpha} \mid \tau \to \sigma$
Contexts	Γ, Δ, Θ	$::=$	$\cdot \mid \Gamma, \alpha \mid \Gamma, x : A$
			$\mid \Gamma, \hat{\alpha} \mid \Gamma, \hat{\alpha} = \tau \mid \Gamma, \blacktriangleright_{\hat{\alpha}}$

Contexts:

$\hat{\alpha}$	**Unsolved** existential
$\hat{\alpha} = \tau$	**Solved** existential
$\blacktriangleright_{\hat{\alpha}}$	Scope marker

Algorithmic contexts

$\hat{\alpha}$	**Unsolved** existential
$\hat{\alpha} = \tau$	**Solved** existential

Contexts are ordered:

- in $(\Gamma, x : A, \Gamma')$, the type A is well-formed under Γ
- in $(\Gamma, \hat{\alpha} = \tau, \Gamma')$, the type τ is well-formed under Γ
- **no multiple declarations:**

$$\hat{\alpha}, \ldots, \hat{\alpha} \quad ✗$$
$$\hat{\alpha}, \ldots, \hat{\alpha} = \tau \quad ✗$$
$$\hat{\alpha} = \tau, \ldots, \hat{\alpha} = \tau \quad ✗$$

Algorithmic typing

Judgments:

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Key invariant: Γ *is extended by* Δ:

$$\Gamma \longrightarrow \Delta$$

Intuition: Δ has at least as much information as Γ

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\dfrac{\dfrac{\vdash () \Leftarrow \quad \dashv}{\vdash \quad \bullet\, () \Rrightarrow \quad \dashv}}{\vdash \quad \bullet\, () \Rrightarrow \quad \dashv} \qquad \Gamma \vdash id \Rightarrow \quad \dashv}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\to\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\vdash () \Leftarrow \quad \dashv}{\vdash \quad \bullet () \Rrightarrow \quad \dashv}}{\vdash \quad \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\, \alpha \to \alpha \dashv \Gamma \qquad \dfrac{\dfrac{\vdash () \Leftarrow \quad \dashv}{\vdash \quad \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash (\forall\alpha.\, \alpha \to \alpha) \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\to\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\vdash () \Leftarrow \quad \dashv}{\Gamma, \hat{\alpha} \vdash \hat{\alpha}\to\hat{\alpha} \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash (\forall\alpha.\,\alpha\to\alpha) \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\to\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}{\Gamma, \hat{\alpha} \vdash \hat{\alpha}\to\hat{\alpha} \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash (\forall\alpha.\,\alpha\to\alpha) \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\to\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\vdots}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}}{\dfrac{\Gamma, \hat{\alpha} \vdash \hat{\alpha}\to\hat{\alpha} \bullet () \Rrightarrow \quad \dashv}{\Gamma \vdash (\forall\alpha.\,\alpha\to\alpha) \bullet () \Rrightarrow \quad \dashv}}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context** Δ

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\to\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\Gamma, \hat\alpha \vdash () \Leftarrow \hat\alpha \dashv}{\Gamma, \hat\alpha \vdash \hat\alpha\to\hat\alpha \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash (\forall\alpha.\,\alpha\to\alpha) \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\to\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}{\Gamma, \hat{\alpha} \vdash \hat{\alpha}\to\hat{\alpha} \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash (\forall\alpha.\,\alpha\to\alpha) \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\to\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}{\Gamma, \hat{\alpha} \vdash \hat{\alpha}\to\hat{\alpha} \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash (\forall\alpha.\,\alpha\to\alpha) \bullet () \Rrightarrow \quad \dashv}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\dfrac{\vdots \quad \vdots}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}}{\dfrac{\Gamma, \hat{\alpha} \vdash \hat{\alpha} \to \hat{\alpha} \bullet () \Rrightarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}{\Gamma \vdash (\forall \alpha.\, \alpha \to \alpha) \bullet () \Rrightarrow \quad \dashv}}$$

$$\dfrac{\Gamma \vdash id \Rightarrow \forall \alpha.\, \alpha \to \alpha \dashv \Gamma \qquad |}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context Δ**

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\rightarrow\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}{\Gamma, \hat{\alpha} \vdash \hat{\alpha}\rightarrow\hat{\alpha} \bullet () \Rrightarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}}{\Gamma \vdash (\forall\alpha.\,\alpha\rightarrow\alpha) \bullet () \Rrightarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}}{\Gamma \vdash id\ () \Rightarrow \quad \dashv}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$

$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$

$$\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta$$

Input context Γ **Output context** Δ

Intuition: Δ has at least as much information as Γ

$$\dfrac{\Gamma \vdash id \Rightarrow \forall\alpha.\,\alpha\to\alpha \dashv \Gamma \qquad \dfrac{\dfrac{\dfrac{\vdots \quad \vdots}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}}{\Gamma, \hat{\alpha} \vdash \hat{\alpha}\to\hat{\alpha} \bullet () \Rrightarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}}{\Gamma \vdash (\forall\alpha.\,\alpha\to\alpha) \bullet () \Rrightarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}}{\Gamma \vdash id\ () \Rightarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}$$

Algorithmic typing

$$\Gamma \vdash e \Leftarrow A \dashv \Delta$$
$$\Gamma \vdash e \Rightarrow A \dashv \Delta$$
$$\Gamma \vdash A \bullet e \mathrel{\Rightarrow\!\!\Rightarrow} C \dashv \Delta$$

Input context Γ **Output context** Δ

Intuition: Δ has at least as much information as Γ

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma, \hat\alpha \vdash () \Leftarrow \hat\alpha \dashv \Gamma, \hat\alpha = 1}}{\Gamma, \hat\alpha \vdash \hat\alpha \to \hat\alpha \bullet () \mathrel{\Rightarrow\!\!\Rightarrow} \hat\alpha \dashv \Gamma, \hat\alpha = 1}}{\Gamma \vdash (\forall \alpha.\, \alpha \to \alpha) \bullet () \mathrel{\Rightarrow\!\!\Rightarrow} \hat\alpha \dashv \Gamma, \hat\alpha = 1}$$

$$\dfrac{\Gamma \vdash id \Rightarrow \forall \alpha.\, \alpha \to \alpha \dashv \Gamma \qquad \Gamma \vdash (\forall \alpha.\, \alpha \to \alpha) \bullet () \mathrel{\Rightarrow\!\!\Rightarrow} \hat\alpha \dashv \Gamma, \hat\alpha = 1}{\Gamma \vdash id\ () \Rightarrow \hat\alpha \dashv \Gamma, \hat\alpha = 1}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$ — A is a subtype of B

$\Gamma \vdash \hat{\alpha} :\overset{<}{=} A \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$

$\Gamma \vdash A \overset{<}{=}: \hat{\alpha} \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\vdash () \Rightarrow \quad \dashv \qquad \dfrac{\dfrac{}{\vdash \quad \overset{<}{=}: \quad \dashv}\ \text{InstRSolve}}{\vdash \quad <: \hat{\alpha} \dashv}}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}\ \text{Sub}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$ — A is a subtype of B

$\Gamma \vdash \hat{\alpha} :\leqq A \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$

$\Gamma \vdash A \leqq: \hat{\alpha} \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\Gamma, \hat{\alpha} \vdash () \Rightarrow \quad \dashv \qquad \dfrac{\dfrac{}{\vdash \quad \leqq: \quad \dashv}\ \text{InstRSolve}}{\vdash \quad <: \hat{\alpha} \dashv}}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}\ \text{Sub}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$	A is a subtype of B
$\Gamma \vdash \hat{\alpha} :\stackrel{<}{=} A \dashv \Delta$	Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$
$\Gamma \vdash A \stackrel{<}{=}: \hat{\alpha} \dashv \Delta$	Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\Gamma, \hat{\alpha} \vdash () \Rightarrow 1 \dashv \Gamma, \hat{\alpha} \qquad \dfrac{\dfrac{}{\ \vdash \quad \stackrel{<}{=}: \quad \dashv\ }\ \text{InstRSolve}}{\ \vdash \quad <: \hat{\alpha} \dashv\ }}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv\ }\ \text{Sub}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$ — A is a subtype of B

$\Gamma \vdash \hat{\alpha} :\overset{<}{=} A \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$

$\Gamma \vdash A \overset{<}{=}: \hat{\alpha} \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\Gamma, \hat{\alpha} \vdash () \Rightarrow 1 \dashv \Gamma, \hat{\alpha} \qquad \dfrac{\dfrac{}{\vdash \quad \overset{<}{=}: \quad \dashv}\ \text{InstRSolve}}{\Gamma, \hat{\alpha} \vdash 1 <: \hat{\alpha} \dashv}}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}\ \text{Sub}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$	A is a subtype of B
$\Gamma \vdash \hat{\alpha} :\stackrel{<}{=} A \dashv \Delta$	Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$
$\Gamma \vdash A \stackrel{<}{=}: \hat{\alpha} \dashv \Delta$	Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\Gamma, \hat{\alpha} \vdash () \Rightarrow 1 \dashv \Gamma, \hat{\alpha} \qquad \dfrac{\overline{\Gamma, \hat{\alpha} \vdash 1 \stackrel{<}{=}: \hat{\alpha} \dashv}\ \text{InstRSolve}}{\Gamma, \hat{\alpha} \vdash 1 <: \hat{\alpha} \dashv}}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}\ \text{Sub}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$	A is a subtype of B
$\Gamma \vdash \hat{\alpha} :\overset{\leq}{=} A \dashv \Delta$	Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$
$\Gamma \vdash A \overset{\leq}{=}: \hat{\alpha} \dashv \Delta$	Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\Gamma, \hat{\alpha} \vdash () \Rightarrow 1 \dashv \Gamma, \hat{\alpha} \qquad \dfrac{\dfrac{\Gamma \vdash 1}{\Gamma, \hat{\alpha} \vdash 1 \overset{\leq}{=}: \hat{\alpha} \dashv}\ \text{InstRSolve}}{\Gamma, \hat{\alpha} \vdash 1 <: \hat{\alpha} \dashv}}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}\ \text{Sub}$$

$$\dfrac{\Gamma \vdash \tau}{\Gamma, \hat{\alpha}, \Gamma' \vdash \tau \overset{\leq}{=}: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = \tau, \Gamma'}\ \text{InstRSolve}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$ — A is a subtype of B

$\Gamma \vdash \hat{\alpha} :\overset{\leq}{=} A \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$

$\Gamma \vdash A \overset{\leq}{=}: \hat{\alpha} \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\Gamma, \hat{\alpha} \vdash () \Rightarrow 1 \dashv \Gamma, \hat{\alpha} \qquad \dfrac{\dfrac{\Gamma \vdash 1}{\Gamma, \hat{\alpha} \vdash 1 \overset{\leq}{=}: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}\,\text{InstRSolve}}{\Gamma, \hat{\alpha} \vdash 1 <: \hat{\alpha} \dashv}}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}\,\text{Sub}$$

$$\dfrac{\Gamma \vdash \tau}{\Gamma, \hat{\alpha}, \Gamma' \vdash \tau \overset{\leq}{=}: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = \tau, \Gamma'}\,\text{InstRSolve}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$	A is a subtype of B
$\Gamma \vdash \hat{\alpha} :\overset{<}{=} A \dashv \Delta$	Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$
$\Gamma \vdash A \overset{<}{=}: \hat{\alpha} \dashv \Delta$	Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\Gamma, \hat{\alpha} \vdash () \Rightarrow 1 \dashv \Gamma, \hat{\alpha} \qquad \dfrac{\dfrac{\Gamma \vdash 1}{\Gamma, \hat{\alpha} \vdash 1 \overset{<}{=}: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}\text{InstRSolve}}{\Gamma, \hat{\alpha} \vdash 1 <: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv}\text{Sub}$$

$$\dfrac{\Gamma \vdash \tau}{\Gamma, \hat{\alpha}, \Gamma' \vdash \tau \overset{<}{=}: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = \tau, \Gamma'}\text{InstRSolve}$$

Instantiation and subtyping judgments

$\Gamma \vdash A <: B \dashv \Delta$ — A is a subtype of B

$\Gamma \vdash \hat{\alpha} :\leqq A \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $\hat{\alpha} <: A$

$\Gamma \vdash A \leqq: \hat{\alpha} \dashv \Delta$ — Instantiate $\hat{\alpha}$ such that $A <: \hat{\alpha}$

$$\dfrac{\Gamma, \hat{\alpha} \vdash () \Rightarrow 1 \dashv \Gamma, \hat{\alpha} \qquad \dfrac{\dfrac{\Gamma \vdash 1}{\Gamma, \hat{\alpha} \vdash 1 \leqq: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}\ \text{InstRSolve}}{\Gamma, \hat{\alpha} \vdash 1 <: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}}{\Gamma, \hat{\alpha} \vdash () \Leftarrow \hat{\alpha} \dashv \Gamma, \hat{\alpha} = 1}\ \text{Sub}$$

$$\dfrac{\Gamma \vdash \tau}{\Gamma, \hat{\alpha}, \Gamma' \vdash \tau \leqq: \hat{\alpha} \dashv \Gamma, \hat{\alpha} = \tau, \Gamma'}\ \text{InstRSolve}$$

Instantiation rules

- Can instantiate a less obvious variable:

$$\frac{}{\hat{\alpha}, \ldots, \hat{\beta} \vdash \hat{\alpha} :\leq \hat{\beta} \dashv \hat{\alpha}, \ldots, \hat{\beta} = \hat{\alpha}} \;\textsf{InstLReach}$$

Can't instantiate $\hat{\alpha} = \hat{\beta}$, because $\hat{\beta}$ not in scope.

Instantiation rules

- Can instantiate a less obvious variable:

$$\frac{}{\hat{\alpha}, \ldots, \hat{\beta} \vdash \hat{\alpha} :\leqq \hat{\beta} \dashv \hat{\alpha}, \ldots, \hat{\beta} = \hat{\alpha}} \;\textsf{InstLReach}$$

Can't instantiate $\hat{\alpha} = \hat{\beta}$, because $\hat{\beta}$ not in scope.

- Might not instantiate at all in output context!

$$\dfrac{\dfrac{}{\hat{\beta}, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} \vdash \hat{\alpha} \leqq: \hat{\beta} \dashv \hat{\beta}, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} = \hat{\beta}} \;\textsf{InstRReach}}{\hat{\beta} \vdash \forall \alpha.\, \alpha \leqq: \hat{\beta} \dashv \hat{\beta}} \;\textsf{InstRAllL}$$

Solves new var. $\hat{\alpha}$, but $\hat{\alpha} = \hat{\beta}$ goes out of scope.
(only happens when instantiating to a **polytype**)

The usual caveat

- "For details, see the paper"

But not too many details

$\boxed{\Gamma \vdash A <: B \dashv \Delta}$ Under input context Γ, type A is a subtype of B, with output context Δ

$$\frac{}{\Gamma[\alpha] \vdash \alpha <: \alpha \dashv \Gamma[\alpha]}\ {<:}\text{Var} \qquad \frac{}{\Gamma \vdash 1 <: 1 \dashv \Gamma}\ {<:}\text{Unit} \qquad \frac{}{\Gamma[\hat\alpha] \vdash \hat\alpha <: \hat\alpha \dashv \Gamma[\hat\alpha]}\ {<:}\text{Exvar}$$

$$\frac{\Gamma \vdash B_1 <: A_1 \dashv \Theta \qquad \Theta \vdash [\Theta]A_2 <: [\Theta]B_2 \dashv \Delta}{\Gamma \vdash A_1 \to A_2 <: B_1 \to B_2 \dashv \Delta}\ {<:}{\to}$$

$$\frac{\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha \vdash [\hat\alpha/\alpha]A <: B \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Theta}{\Gamma \vdash \forall\alpha.\, A <: B \dashv \Delta}\ {<:}\forall\text{L} \qquad \frac{\Gamma, \alpha \vdash A <: B \dashv \Delta, \alpha, \Theta}{\Gamma \vdash A <: \forall\alpha.\, B \dashv \Delta}\ {<:}\forall\text{R}$$

$$\frac{\hat\alpha \notin FV(A) \qquad \Gamma[\hat\alpha] \vdash \hat\alpha :\leqq A \dashv \Delta}{\Gamma[\hat\alpha] \vdash \hat\alpha <: A \dashv \Delta}\ {<:}\text{InstantiateL} \qquad \frac{\hat\alpha \notin FV(A) \qquad \Gamma[\hat\alpha] \vdash A \leqq: \hat\alpha \dashv \Delta}{\Gamma[\hat\alpha] \vdash A <: \hat\alpha \dashv \Delta}\ {<:}\text{InstantiateR}$$

Figure 9. Algorithmic subtyping

$\boxed{\Gamma \vdash \hat\alpha :\leqq A \dashv \Delta}$ Under input context Γ, instantiate α̂ such that α̂ <: A, with output context Δ

$$\frac{\Gamma \vdash \tau}{\Gamma, \hat\alpha, \Gamma' \vdash \hat\alpha :\leqq \tau \dashv \Gamma, \hat\alpha = \tau, \Gamma'}\ \text{InstLSolve} \qquad \frac{}{\Gamma[\hat\alpha][\hat\beta] \vdash \hat\alpha :\leqq \hat\beta \dashv \Gamma[\hat\alpha][\hat\beta = \hat\alpha]}\ \text{InstLReach}$$

$$\frac{\Gamma[\hat\alpha_2, \hat\alpha_1, \hat\alpha = \hat\alpha_1 \to \hat\alpha_2] \vdash A_1 \leqq: \hat\alpha_1 \dashv \Theta \qquad \Theta \vdash \hat\alpha_2 :\leqq [\Theta]A_2 \dashv \Delta}{\Gamma[\hat\alpha] \vdash \hat\alpha :\leqq A_1 \to A_2 \dashv \Delta}\ \text{InstLArr} \qquad \frac{\Gamma[\hat\alpha], \beta \vdash \hat\alpha :\leqq B \dashv \Delta, \beta, \Delta'}{\Gamma[\hat\alpha] \vdash \hat\alpha :\leqq \forall\beta.\, B \dashv \Delta}\ \text{InstLAllR}$$

$\boxed{\Gamma \vdash A \leqq: \hat\alpha \dashv \Delta}$ Under input context Γ, instantiate α̂ such that A <: α̂, with output context Δ

$$\frac{\Gamma \vdash \tau}{\Gamma, \hat\alpha, \Gamma' \vdash \tau \leqq: \hat\alpha \dashv \Gamma, \hat\alpha = \tau, \Gamma'}\ \text{InstRSolve} \qquad \frac{}{\Gamma[\hat\alpha][\hat\beta] \vdash \hat\beta \leqq: \hat\alpha \dashv \Gamma[\hat\alpha][\hat\beta = \hat\alpha]}\ \text{InstRReach}$$

$$\frac{\Gamma[\hat\alpha_2, \hat\alpha_1, \hat\alpha = \hat\alpha_1 \to \hat\alpha_2] \vdash \hat\alpha_1 :\leqq A_1 \dashv \Theta \qquad \Theta \vdash [\Theta]A_2 \leqq: \hat\alpha_2 \dashv \Delta}{\Gamma[\hat\alpha] \vdash A_1 \to A_2 \leqq: \hat\alpha \dashv \Delta}\ \text{InstRArr} \qquad \frac{\Gamma[\hat\alpha], \blacktriangleright_{\hat\beta}, \hat\beta \vdash [\hat\beta/\beta]B \leqq: \hat\alpha \dashv \Delta, \blacktriangleright_{\hat\beta}, \Delta'}{\Gamma[\hat\alpha] \vdash \forall\beta.\, B \leqq: \hat\alpha \dashv \Delta}\ \text{InstRAllL}$$

Figure 10. Instantiation

$\boxed{\Gamma \vdash e \Leftarrow A \dashv \Delta}$ Under input context Γ, e checks against input type A, with output context Δ

$\boxed{\Gamma \vdash e \Rightarrow A \dashv \Delta}$ Under input context Γ, e synthesizes output type A, with output context Δ

$\boxed{\Gamma \vdash A \bullet e \Rrightarrow C \dashv \Delta}$ Under input context Γ, applying a function of type A to e synthesizes type C, with output context Δ

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x \Rightarrow A \dashv \Gamma}\ \text{Var} \qquad \frac{\Gamma \vdash e \Rightarrow A \dashv \Theta \qquad \Theta \vdash [\Theta]A <: [\Theta]B \dashv \Delta}{\Gamma \vdash e \Leftarrow B \dashv \Delta}\ \text{Sub} \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash e \Leftarrow A \dashv \Delta}{\Gamma \vdash (e : A) \Rightarrow A \dashv \Delta}\ \text{Anno}$$

$$\frac{}{\Gamma \vdash () \Leftarrow 1 \dashv \Gamma}\ 1\text{I} \qquad \frac{}{\Gamma \vdash () \Rightarrow 1 \dashv \Gamma}\ 1\text{I}{\Rightarrow} \qquad \frac{\Gamma, \alpha \vdash e \Leftarrow A \dashv \Delta, \alpha, \Theta}{\Gamma \vdash e \Leftarrow \forall\alpha.\, A \dashv \Delta}\ \forall\text{I} \qquad \frac{\Gamma, \hat\alpha \vdash [\hat\alpha/\alpha]A \bullet e \Rrightarrow C \dashv \Delta}{\Gamma \vdash \forall\alpha.\, A \bullet e \Rrightarrow C \dashv \Delta}\ \forall\text{App}$$

$$\frac{\Gamma, x : A \vdash e \Leftarrow B \dashv \Delta, x : A, \Theta}{\Gamma \vdash \lambda x.\, e \Leftarrow A \to B \dashv \Delta}\ {\to}\text{I} \qquad \frac{\Gamma, \hat\alpha, \hat\beta, x : \hat\alpha \vdash e \Leftarrow \hat\beta \dashv \Delta, x : \hat\alpha, \Theta}{\Gamma \vdash \lambda x.\, e \Rightarrow \hat\alpha \to \hat\beta \dashv \Delta}\ {\to}\text{I}{\Rightarrow} \qquad \frac{\Gamma \vdash e_1 \Rightarrow A \dashv \Theta \qquad \Theta \vdash [\Theta]A \bullet e_2 \Rrightarrow C \dashv \Delta}{\Gamma \vdash e_1\, e_2 \Rightarrow C \dashv \Delta}\ {\to}\text{E}$$

$$\frac{\Gamma[\hat\alpha_2, \hat\alpha_1, \hat\alpha = \hat\alpha_1 \to \hat\alpha_2] \vdash e \Leftarrow \hat\alpha_1 \dashv \Delta}{\Gamma[\hat\alpha] \vdash \hat\alpha \bullet e \Rrightarrow \hat\alpha_2 \dashv \Delta}\ \hat\alpha\text{App} \qquad \frac{\Gamma \vdash e \Leftarrow A \dashv \Delta}{\Gamma \vdash A \to C \bullet e \Rrightarrow C \dashv \Delta}\ {\to}\text{App}$$

Figure 11. Algorithmic typing

Easy!

http://semantic-domain.blogspot.de/2013/04/thanks-to-olle-fredriksson.html

Wednesday, April 10, 2013

Thanks to Olle Fredriksson

Joshua and I thought our new typechecking algorithm for higher-rank polymorphism was easy to implement, but I have to admit that we weren't 100% sure about this, since we've been immersed in this work for months.

However, Olle Fredriksson wrote us within one day of the draft going online with a link to his Haskell implementation. This was *immensely* reassuring, since (a) he implemented it so quickly, and (b) he didn't need to ask us how to implement it -- this means we didn't leave anything important out in the description!

In fact, Olle is technically the first person to implement the algorithm in the paper, since both Joshua and I have actually implemented variants of this algorithm for different languages. I'll try to write a small ML implementation to go along with Olle's Haskell code in the next few weeks. If you want to beat me to it, though, I have no objections!

Posted by Neel Krishnaswami at 12:41 PM

The algorithm is

- **Decidable** (count $\forall$s; unsolved $\hat{\alpha}$s; "contextual size")

The algorithm is

- **Decidable** (count $\forall$s; unsolved $\hat{\alpha}$s; "contextual size")
- **Sound**

  If $\Delta \longrightarrow \Omega$ and $\hat{\alpha} \notin FV(B)$:

  (1) If $\Gamma \vdash \hat{\alpha} :\stackrel{\leq}{=} B \dashv \Delta$ then $[\Omega]\Delta \vdash [\Omega]\hat{\alpha} \leq [\Omega]B$.

  (2) If $\Gamma \vdash B \stackrel{\leq}{=}: \hat{\alpha} \dashv \Delta$ then $[\Omega]\Delta \vdash [\Omega]B \leq [\Omega]\hat{\alpha}$.

  (and similar results for subtyping and typing)

The algorithm is

- **Decidable** (count $\forall$s; unsolved $\hat{\alpha}$s; “contextual size”)
- **Sound**

  If $\Delta \longrightarrow \Omega$ and $\hat{\alpha} \notin FV(B)$:

  (1) If $\Gamma \vdash \hat{\alpha} :\overset{\leq}{=} B \dashv \Delta$ then $[\Omega]\Delta \vdash [\Omega]\hat{\alpha} \leq [\Omega]B$.
  (2) If $\Gamma \vdash B \overset{\leq}{=}: \hat{\alpha} \dashv \Delta$ then $[\Omega]\Delta \vdash [\Omega]B \leq [\Omega]\hat{\alpha}$.

  (and similar results for subtyping and typing)

- **Complete**

  If $\Gamma \longrightarrow \Omega$ and $[\Omega]\Gamma \vdash [\Omega]A \leq [\Omega]B$
  then there exists appropriate Δ such that
  $\Gamma \vdash [\Gamma]A <: [\Gamma]B \dashv \Delta$.

  (glossing over some details)

  (and similar results for instantiation and typing)

Related work

- Our approach:

  (1) preserves types under η-reduction (and expansion)
  (2) keeps the standard types of System F

- ... but does **not**:

  (3) allow impredicative instantiation of $\forall$

- Undecidability results prevent achieving all three
- ML^F (Le Botlan–Rémy–Yakobowski) drops (2) and gets (3)
- FPH (Vytiniotis et al. 2008) and HML (Leijen 2009) drop (1) and get (3)
- Peyton Jones et al. (2007) have (1) and (2), like us

Related work

- This work started as an attempt to extend "Greedy bidirectional polymorphism" (Dunfield 2009) with type-level computation
- We found several problems…
  - the most embarrassing: a measure on judgments used to "prove" decidability was **not even transitive**
- The centrality of **ordered contexts** (suggested by Brigitte Pientka) lives on in this paper
- Other notable uses of ordered contexts:
  - Gundry et al. (2010), as a new take on Damas-Milner
  - Miller (1992), as mixed-prefix unification

Summary

- Declarative bidirectional system:
  - annotatability/soundness wrt System F
  - specifies when the algorithm should succeed
- Bidirectional typing algorithm:
  - based on instantiation judgment
  - inst., subtyping, typing rules fit on a page
  - sound and **complete** wrt declarative system
  - **easy** to implement:
    done in **one day** from the paper alone
- **Future work:**
  - Extend to GADTs, type-level computation, . . .
  - Support for “manual” impredicative instantiation

Conclusion:

Typing higher-rank polymorphism can be
complete and **easy**
using our bidirectional approach.

Paper and full proofs:

arxiv.org/abs/1306.6032

Thank you!

Shameless advertisement:
I'm looking for a research or teaching job in Canada. 

Conclusion:

Typing higher-rank polymorphism can be
complete and **easy**
using our bidirectional approach.

Paper and full proofs:

`arxiv.org/abs/1306.6032`

Thank you!

Shameless advertisement:
I'm looking for a research or teaching job in Canada. 

Algorithmic subtyping

- Recall two of the declarative rules:

$$\frac{\Psi \vdash \tau \qquad \Psi \vdash [\tau/\alpha]A \leq B}{\Psi \vdash \forall\alpha.\, A \leq B} \leq\!\forall\mathsf{L} \qquad \frac{\Psi, \beta \vdash A \leq B}{\Psi \vdash A \leq \forall\beta.\, B} \leq\!\forall\mathsf{R}$$

- Algorithmic versions:

$$\frac{\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} \vdash [\hat{\alpha}/\alpha]A <: B \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta}{\Gamma \vdash \forall\alpha.\, A <: B \dashv \Delta} <:\!\forall\mathsf{L}$$

$$\frac{\Gamma, \alpha \vdash A <: B \dashv \Delta, \alpha, \Theta}{\Gamma \vdash A <: \forall\alpha.\, B \dashv \Delta} <:\!\forall\mathsf{R}$$