Type inference

Gabriel Scherer

2025-10-22

MPRI

Intro

What nobody wants

```
let rec length (type a) (li : a list) : int =
  match li with
  | [] -> 0
  | (x : a) :: (xs : a list) ->
    1 + length (type a) xs
```

```
let rec length (type a) (li : a list) : int =
  match li with
  | [] -> 0
  | (x : a) :: (xs : a list) ->
    1 + length (type a) xs
```

Can we maybe have type-checking, without writing so much types?

What nobody wants

```
let rec length (type a) (li : a list) : int =
  match li with
  | [] -> 0
  | (x : a) :: (xs : a list) ->
    1 + length (type a) xs
```

Can we maybe have type-checking, without writing so much types?

```
let rec length li =
  match li with
  | [] -> 0
  | x :: xs ->
    1 + length xs
```

Type inference

Guessing the types.

Sounds boring, but very useful!

Meta-theory: mathematics of usability.

Generalities

Type inference, formally

From a program t in some typing context Γ,
can we **guess** a type A and a derivation of $\Gamma \vdash t : A$?

Distinction: decidability

What does it mean that a type system is *decidable* ? (STLC, System F, CoC, ETT…)

Distinction: decidability

What does it mean that a type system is *decidable*? (STLC, System F, CoC, ETT...)

$$\Gamma \vdash ? : A$$

Distinction: decidability

What does it mean that a type system is *decidable* ? (STLC, System F, CoC, ETT…)

$\Gamma \vdash ? : A$ Inhabitation

Distinction: decidability

What does it mean that a type system is *decidable* ? (STLC, System F, CoC, ETT…)

$\Gamma \vdash ? : A$	Inhabitation
$\Gamma \vdash ? : 0$	Context (in)consistency

Distinction: decidability

What does it mean that a type system is *decidable*? (STLC, System F, CoC, ETT...)

$\Gamma \vdash ? : A$	Inhabitation
$\Gamma \vdash ? : 0$	Context (in)consistency
$\Gamma \vdash t : ?$	Typability ; Type inference

Distinction: decidability

What does it mean that a type system is *decidable* ? (STLC, System F, CoC, ETT...)

$\Gamma \vdash ? : A$	Inhabitation
$\Gamma \vdash ? : 0$	Context (in)consistency
$\Gamma \vdash t : ?$	Typability ; Type inference
$? \vdash t : A$	Context inference

Distinction: decidability

What does it mean that a type system is *decidable* ? (STLC, System F, CoC, ETT...)

$\Gamma \vdash ? : A$	Inhabitation
$\Gamma \vdash ? : 0$	Context (in)consistency
$\Gamma \vdash t : ?$	Typability ; Type inference
$? \vdash t : A$	Context inference
$? \vdash t : ?$	Typing inference

Distinction: decidability

What does it mean that a type system is *decidable* ? (STLC, System F, CoC, ETT...)

$\Gamma \vdash ? : A$	Inhabitation
$\Gamma \vdash ? : 0$	Context (in)consistency
$\Gamma \vdash t : ?$	Typability ; Type inference
$? \vdash t : A$	Context inference
$? \vdash t : ?$	Typing inference
$\Gamma \vdash t : A$	Decidability

Distinction: decidability

What does it mean that a type system is *decidable*? (STLC, System F, CoC, ETT...)

$\Gamma \vdash ? : A$	Inhabitation
$\Gamma \vdash ? : 0$	Context (in)consistency
$\Gamma \vdash t : ?$	Typability ; Type inference
$? \vdash t : A$	Context inference
$? \vdash t : ?$	Typing inference
$\Gamma \vdash t : A$	Decidability

Remark: the game also has a relational version:

$$\Gamma \vdash_1 t : A \quad \Longrightarrow \quad \Gamma \vdash_2 t : A$$
$$\Gamma \vdash_1 ? : A \quad \Longrightarrow \quad \Gamma \vdash_2 ? : A$$
$$\Gamma \vdash_1 t : ? \quad \Longrightarrow \quad \Gamma \vdash_2 t : ?$$

Implicit vs. Explicit syntax

Implicitly typed syntax (Curry-style):

$$t ::= x, y, z \mid \lambda x.\, t \mid t\; u$$

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

$$\emptyset \vdash \lambda x.\, x : ?$$

Implicit vs. Explicit syntax

Implicitly typed syntax (Curry-style):

$$t ::= x, y, z \mid \lambda x.\, t \mid t\, u$$

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

$$\emptyset \vdash \lambda x.\, x : ?$$

Explicitly typed syntax (Church-style):

$$t ::= x, y, z \mid \lambda x{:}A.\, t \mid t\, u$$

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

$$\emptyset \vdash \lambda x{:}A.\, x : ?$$

Implicit vs. Explicit syntax

Implicitly typed syntax (Curry-style):

$$t ::= x, y, z \mid \lambda x.\, t \mid t\, u$$

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

$$\emptyset \vdash \lambda x.\, x : ?$$

Explicitly typed syntax (Church-style):

$$t ::= x, y, z \mid \lambda x{:}A.\, t \mid t\, u$$

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

$$\emptyset \vdash \lambda x{:}A.\, x : ?$$

Remark:

$$\frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Implicit vs. Explicit syntax

Implicitly typed syntax (Curry-style):

$$t ::= x, y, z \mid \lambda x.\, t \mid t\, u$$

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

$$\emptyset \vdash \lambda x.\, x : ?$$

Explicitly typed syntax (Church-style):

$$t ::= x, y, z \mid \lambda x{:}A.\, t \mid t\, u$$

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

$$\emptyset \vdash \lambda x{:}A.\, x : ?$$

Remark:

$$\frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Explicit syntax: one-to-one correspondence with derivations.

Implicitly typed syntax (Curry-style):

$$t ::= x, y, z \mid \lambda x.\, t \mid t\, u$$

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

$$\emptyset \vdash \lambda x.\, x : ?$$

Explicitly typed syntax (Church-style):

$$t ::= x, y, z \mid \lambda x{:}A.\, t \mid t\, u$$

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

$$\emptyset \vdash \lambda x{:}A.\, x : ?$$

Remark:

$$\frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Explicit syntax: one-to-one correspondence with derivations.

(*Syntax-directed*: at most one typing rule per term.)

Implicit vs. Explicit syntax

Implicitly typed syntax (Curry-style):

$$t ::= x, y, z \mid \lambda x.\, t \mid t\ u$$

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

$$\emptyset \vdash \lambda x.\, x : ?$$

Explicitly typed syntax (Church-style):

$$t ::= x, y, z \mid \lambda x{:}A.\, t \mid t\ u$$

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

$$\emptyset \vdash \lambda x{:}A.\, x : ?$$

Remark:

$$\frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B}$$

Explicit syntax: one-to-one correspondence with derivations.

(*Syntax-directed*: at most one typing rule per term.)

Inference: from implicit to explicit.

Implicit vs. Explicit: System F

Implicit:

$$t ::= x, y, z \mid \lambda x.\, t \mid t\, u$$

$$\frac{\Gamma, \alpha \vdash t : A}{\Gamma \vdash t : \forall \alpha.\, A}$$

$$\frac{\Gamma \vdash t : \forall \alpha.\, A \quad \Gamma \vdash B}{\Gamma \vdash t : A[\alpha := B]}$$

Implicit vs. Explicit: System F

Implicit:

$$t ::= x, y, z \mid \lambda x.\, t \mid t\ u$$

$$\frac{\Gamma, \alpha \vdash t : A}{\Gamma \vdash t : \forall \alpha.\, A}$$

$$\frac{\Gamma \vdash t : \forall \alpha.\, A \quad \Gamma \vdash B}{\Gamma \vdash t : A[\alpha := B]}$$

Explicit:

$$t ::= x, y, z \mid \lambda x{:}A.\, t \mid t\ u \mid \Lambda \alpha.\, t \mid t\ [A]$$

$$\frac{\Gamma, \alpha \vdash t : A}{\Gamma \vdash \Lambda \alpha.\, t : \forall \alpha.\, A}$$

$$\frac{\Gamma \vdash t : \forall \alpha.\, A \quad \Gamma \vdash B}{\Gamma \vdash t\ [B] : A[\alpha := B]}$$

Implicit vs. Explicit: System F

Implicit:

$$t ::= x, y, z \mid \lambda x.\, t \mid t\, u$$

$$\frac{\Gamma, \alpha \vdash t : A}{\Gamma \vdash t : \forall \alpha.\, A}$$

$$\frac{\Gamma \vdash t : \forall \alpha.\, A \quad \Gamma \vdash B}{\Gamma \vdash t : A[\alpha := B]}$$

Explicit:

$$t ::= x, y, z \mid \lambda x{:}A.\, t \mid t\, u \mid \Lambda \alpha.\, t \mid t\, [A]$$

$$\frac{\Gamma, \alpha \vdash t : A}{\Gamma \vdash \Lambda \alpha.\, t : \forall \alpha.\, A}$$

$$\frac{\Gamma \vdash t : \forall \alpha.\, A \quad \Gamma \vdash B}{\Gamma \vdash t\, [B] : A[\alpha := B]}$$

Implicit polymorphism is not syntax-directed.

Inhabitation and type inference are undecidable.

Not the end!

Type inference is undecidable for full System F.

Two avenues of progress:

- require some type annotations from the user (more explicit syntax)
- move to a less powerful type system

Tweak the language to make inference possible.

In this course, two common approaches:

1. Bidirectional type inference (for STLC)
2. Hindley-Damas-Milner type inference (for ML)

Two basic proof steps to pass this course: **induction** and **inversion**.

Inversion on inference rules: eliminate rules that do not match the goal.

Example:

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B} \qquad v ::= x \mid \lambda x.\, t$$

Prove that $\emptyset \vdash v : A \to B$ implies $x{:}A \vdash t : B$ for some x, t such that $v = \lambda x.\, t$.

Bidirectional type inference for STLC

Idea

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

Idea: we cannot easily *infer* $\Gamma \vdash \lambda x.\, t : ?$ but we can easily *check* $\Gamma \vdash \lambda x.\, t : A \to B$.

Idea

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

Idea: we cannot easily *infer* $\Gamma \vdash \lambda x.\, t : ?$ but we can easily *check* $\Gamma \vdash \lambda x.\, t : A \to B$.

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \;\;:\;\; A} \qquad \frac{\Gamma, x{:}A \vdash t \;\;:\;\; B}{\Gamma \vdash \lambda x.\, t \;\;:\;\; (A \to B)} \qquad \frac{\Gamma \vdash t \;\;:\;\; (A \to B) \quad \Gamma \vdash u \;\;:\;\; A}{\Gamma \vdash t\, u \;\;:\;\; B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \;:\; A} \qquad \frac{\Gamma, x{:}A \vdash t \;:\; B}{\Gamma \vdash \lambda x.\, t \;:\; (A \to B)} \qquad \frac{\Gamma \vdash t \;:\; (A \to B) \quad \Gamma \vdash u \;:\; A}{\Gamma \vdash t\, u \;:\; B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \ : \ B}{\Gamma \vdash \lambda x.\, t \ : \ (A \to B)} \qquad \frac{\Gamma \vdash t \ : \ (A \to B) \quad \Gamma \vdash u \ : \ A}{\Gamma \vdash t\, u \ : \ B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \ : \ B}{\Gamma \vdash \lambda x.\, t \ : \ (A \to B)} \qquad \frac{\Gamma \vdash t \ : \ (A \to B) \quad \Gamma \vdash u \ : \ A}{\Gamma \vdash t\, u \ : \ B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \;\;:\;\; B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \;\;:\;\; (A \to B) \quad \Gamma \vdash u \;\;:\;\; A}{\Gamma \vdash t\, u \;\;:\;\; B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \ : \ B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \ : \ (A \to B) \quad \Gamma \vdash u \ : \ A}{\Gamma \vdash t\, u \ : \ B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \;:\; (A \to B) \quad \Gamma \vdash u \;:\; A}{\Gamma \vdash t\, u \;:\; B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \;:\; (A \to B) \quad \Gamma \vdash u \;:\; A}{\Gamma \vdash t\, u \;:\; B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \Rightarrow (A \to B) \quad \Gamma \vdash u \;:\; A}{\Gamma \vdash t\, u \;:\; B}$$

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \Rightarrow (A \to B) \quad \Gamma \vdash u \;:\; A}{\Gamma \vdash t\, u \;:\; B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \Rightarrow (A \to B) \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \Rightarrow (A \to B) \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Test: can we prove $\emptyset \vdash \lambda x.\, x \Leftarrow A \to A$?

Re-inventing bidirectional inference

$$\frac{}{\Gamma, x{:}A \vdash x : A} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B}$$

Separate $\Gamma \vdash t : A$ into two judgments:

Check: $\Gamma \vdash t \Leftarrow A$ Infer: $\Gamma \vdash t \Rightarrow A$

$$\frac{}{\Gamma, x{:}A \vdash x \Rightarrow A} \qquad \frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow (A \to B)} \qquad \frac{\Gamma \vdash t \Rightarrow (A \to B) \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Test: can we prove $\emptyset \vdash \lambda x.\, x \Leftarrow A \to A$?

$$\frac{\Gamma \vdash t \Rightarrow A}{\Gamma \vdash t \Leftarrow A}$$

Meta-theory

$$\frac{\Gamma \vdash t \Rightarrow A}{\Gamma \vdash t \Leftarrow A} \qquad \frac{}{\Gamma, x{:}A \vdash x \Rightarrow A}$$

$$\frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow A \to B} \qquad \frac{\Gamma \vdash t \Rightarrow A \to B \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Soundness: $\Gamma \vdash t \Leftarrow A \quad \vee \quad \Gamma \vdash t \Rightarrow A \qquad \Longrightarrow \qquad \Gamma \vdash t : A$?

Meta-theory

$$\frac{\Gamma \vdash t \Rightarrow A}{\Gamma \vdash t \Leftarrow A} \qquad \frac{}{\Gamma, x{:}A \vdash x \Rightarrow A}$$

$$\frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow A \to B} \qquad \frac{\Gamma \vdash t \Rightarrow A \to B \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Soundness: $\Gamma \vdash t \Leftarrow A \quad \vee \quad \Gamma \vdash t \Rightarrow A \quad \Longrightarrow \quad \Gamma \vdash t : A \quad ?$

Completeness: $\Gamma \vdash t : A \quad \Longrightarrow \quad \Gamma \vdash t \Leftarrow A \quad ?$

Meta-theory

$$\frac{\Gamma \vdash t \Rightarrow A}{\Gamma \vdash t \Leftarrow A} \qquad \frac{}{\Gamma, x{:}A \vdash x \Rightarrow A}$$

$$\frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow A \to B} \qquad \frac{\Gamma \vdash t \Rightarrow A \to B \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Soundness: $\Gamma \vdash t \Leftarrow A \quad \vee \quad \Gamma \vdash t \Rightarrow A \quad \Longrightarrow \quad \Gamma \vdash t : A$?

Completeness: $\Gamma \vdash t : A \quad \Longrightarrow \quad \Gamma \vdash t \Leftarrow A$?

Problem:

Meta-theory

$$\frac{\Gamma \vdash t \Rightarrow A}{\Gamma \vdash t \Leftarrow A} \qquad \frac{}{\Gamma, x{:}A \vdash x \Rightarrow A}$$

$$\frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow A \to B} \qquad \frac{\Gamma \vdash t \Rightarrow A \to B \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Soundness: $\Gamma \vdash t \Leftarrow A \quad \vee \quad \Gamma \vdash t \Rightarrow A \qquad \Longrightarrow \qquad \Gamma \vdash t : A$?

Completeness: $\Gamma \vdash t : A \qquad \Longrightarrow \qquad \Gamma \vdash t \Leftarrow A$?

Problem: $(\lambda x.\, t)\; u$

Solution:

Meta-theory

$$\frac{\Gamma \vdash t \Rightarrow A}{\Gamma \vdash t \Leftarrow A} \qquad \frac{}{\Gamma, x{:}A \vdash x \Rightarrow A}$$

$$\frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow A \to B} \qquad \frac{\Gamma \vdash t \Rightarrow A \to B \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Soundness: $\Gamma \vdash t \Leftarrow A \quad \vee \quad \Gamma \vdash t \Rightarrow A \quad \Longrightarrow \quad \Gamma \vdash t : A$?

Completeness: $\Gamma \vdash t : A \quad \Longrightarrow \quad \Gamma \vdash t \Leftarrow A$?

Problem: $(\lambda x.\, t)\; u$

Solution: $t ::= \ldots \mid (t : A)$

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

$$\frac{\Gamma \vdash t \Rightarrow A}{\Gamma \vdash t \Leftarrow A} \qquad \frac{}{\Gamma, x{:}A \vdash x \Rightarrow A}$$

$$\frac{\Gamma, x{:}A \vdash t \Leftarrow B}{\Gamma \vdash \lambda x.\, t \Leftarrow A \to B} \qquad \frac{\Gamma \vdash t \Rightarrow A \to B \quad \Gamma \vdash u \Leftarrow A}{\Gamma \vdash t\, u \Rightarrow B}$$

Soundness: $\Gamma \vdash t \Leftarrow A \quad \vee \quad \Gamma \vdash t \Rightarrow A \quad \Longrightarrow \quad \Gamma \vdash t : A$?

Completeness: $\Gamma \vdash t : A \quad \Longrightarrow \quad \Gamma \vdash t \Leftarrow A$?

Problem: $(\lambda x.\, t)\, u$

Solution: $t \mathrel{::=} \ldots \mid (t : A)$ $\qquad \dfrac{\Gamma \vdash t \Leftarrow A}{\Gamma \vdash (t : A) \Rightarrow A}$

Completeness: $\Gamma \vdash t : A \quad \Longrightarrow \quad \exists t',\quad \lfloor t' \rfloor = t \quad \wedge \quad \Gamma \vdash t' \Leftarrow A$

Bidirectional type inference

Bidirectional typing works *really well* for $\beta\eta$-normal forms.

$$n ::= x \mid n\,t \mid (t : A)$$

$$t ::= \lambda x.\, t \mid n$$

For actual programs, it requires adding some annotations.

Bidirectional type inference

Bidirectional typing works *really well* for $\beta\eta$-normal forms.

$$n ::= x \mid n\, t \mid (t : A)$$
$$t ::= \lambda x.\, t \mid n$$

For actual programs, it requires adding some annotations.

Simple to implement and extend, fairly intuitive, a good first step.

Bidirectional typing works *really well* for $\beta\eta$-normal forms.

$$n ::= x \mid n\,t \mid (t : A)$$
$$t ::= \lambda x.\, t \mid n$$

For actual programs, it requires adding some annotations.

Simple to implement and extend, fairly intuitive, a good first step.

Problems:

1. Simple program transformations break typability.

$$t\,u \qquad\qquad \Longrightarrow \qquad\qquad \text{let app} = \lambda x.\, \lambda y.\, x\ y \text{ in app}\ t\ u$$

2. Non-canonical propagation choice, may not be what you want.

Bidirectional type inference

Bidirectional typing works *really well* for $\beta\eta$-normal forms.

$$n ::= x \mid n\,t \mid (t : A)$$
$$t ::= \lambda x.\, t \mid n$$

For actual programs, it requires adding some annotations.

Simple to implement and extend, fairly intuitive, a good first step.

Problems:

1. Simple program transformations break typability.

$$t\ u \qquad\qquad \Longrightarrow \qquad\qquad \text{let app} = \lambda x.\, \lambda y.\, x\ y \text{ in app}\ t\ u$$

2. Non-canonical propagation choice, may not be what you want.

For System F and beyond: see [Dunfield and Krishnaswami 2021].

Hindley-Damas-Milner type inference for ML

Idea (1)

$$A ::= X, Y, Z \mid A \to B \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B}$$

Idea: if we read this rule in mode $\Gamma \vdash t : ?$ then A is *unknown*

Idea (1)

$$A ::= X, Y, Z \mid A \to B$$

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

Idea: if we read this rule in mode $\Gamma \vdash t : ?$ then A is *unknown*

Unknown type variables $A ::= \ldots. \mid a_?, b_?, c_?$

$$\frac{\Gamma, x{:}a_? \vdash_{\text{inf}} t : B}{\Gamma \vdash_{\text{inf}} \lambda x.\, t : a_? \to B}$$

$$A ::= X, Y, Z \mid A \to B \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x.\, t : A \to B}$$

Idea: if we read this rule in mode $\Gamma \vdash t : ?$ then A is *unknown*

Unknown type variables $A ::= \ldots. \mid a_?, b_?, c_?$

$$\frac{\Gamma, x{:}a_? \vdash_{\text{inf}} t : B}{\Gamma \vdash_{\text{inf}} \lambda x.\, t : a_? \to B}$$

To express the application rule in this style, we use unification constraints $A \stackrel{?}{=} B$:

$$\frac{\Gamma \vdash_{\text{inf}} t : A \qquad \Gamma \vdash_{\text{inf}} u : B \qquad A \stackrel{?}{=} (B \to c_?)}{\Gamma \vdash_{\text{inf}} t\, u : c_?}$$

Idea (2)

$$\frac{}{\Gamma, x{:}A \vdash_{\text{inf}} x : A} \qquad \frac{\Gamma, x{:}a_? \vdash_{\text{inf}} t : B}{\Gamma \vdash_{\text{inf}} \lambda x.\, t : a_? \to B} \qquad \frac{\Gamma \vdash_{\text{inf}} t : A \quad \Gamma \vdash_{\text{inf}} u : B \quad A \stackrel{?}{=} (B \to c_?)}{\Gamma \vdash_{\text{inf}} t\, u : c_?}$$

Typing $\Gamma \vdash_{\text{inf}} t : ?$ gives:

- a type A
- mentioning some unknowns $\overline{a_?}$
- and a collection of unification constraints $A \stackrel{?}{=} B$

A *system of equations.*

If a *solution* γ exists, mapping unknowns to known types, we have $\gamma(\Gamma) \vdash t : \gamma(A)$.

Example: $\emptyset \vdash \lambda x.\, \lambda y.\, x\, y : ?$

Idea (2)

$$\frac{}{\Gamma, x{:}A \vdash_{\text{inf}} x : A} \qquad \frac{\Gamma, x{:}a_? \vdash_{\text{inf}} t : B}{\Gamma \vdash_{\text{inf}} \lambda x.\, t : a_? \to B} \qquad \frac{\Gamma \vdash_{\text{inf}} t : A \quad \Gamma \vdash_{\text{inf}} u : B \quad A \stackrel{?}{=} (B \to c_?)}{\Gamma \vdash_{\text{inf}} t\, u : c_?}$$

Typing $\Gamma \vdash_{\text{inf}} t : ?$ gives:

- a type A
- mentioning some unknowns $\overline{a_?}$
- and a collection of unification constraints $A \stackrel{?}{=} B$

A *system of equations.*

If a *solution* γ exists, mapping unknowns to known types, we have $\gamma(\Gamma) \vdash t : \gamma(A)$.

Example: $\emptyset \vdash \lambda x.\, \lambda y.\, x\, y : ?$ $\implies$ $a_? \to b_? \to c_?$ with $a_? = (b_? \to c_?)$

$$\frac{}{\Gamma, x{:}A \vdash_{\text{inf}} x : A} \qquad \frac{\Gamma, x{:}a_? \vdash_{\text{inf}} t : B}{\Gamma \vdash_{\text{inf}} \lambda x.\, t : a_? \to B} \qquad \frac{\Gamma \vdash_{\text{inf}} t : A \quad \Gamma \vdash_{\text{inf}} u : B \quad A \stackrel{?}{=} (B \to c_?)}{\Gamma \vdash_{\text{inf}} t\, u : c_?}$$

Typing $\Gamma \vdash_{\text{inf}} t : ?$ gives:

- a type A
- mentioning some unknowns $\overline{a_?}$
- and a collection of unification constraints $A \stackrel{?}{=} B$

A *system of equations.*

If a *solution* γ exists, mapping unknowns to known types, we have $\gamma(\Gamma) \vdash t : \gamma(A)$.

Example: $\emptyset \vdash \lambda x.\, \lambda y.\, x\, y : ?$ $\implies$ $a_? \to b_? \to c_?$ with $a_? = (b_? \to c_?)$

that is, $(b_? \to c_?) \to (b_? \to c_?)$

Meta-theory

Define a language of *constraints*: $C ::= \mathrm{True} \mid \mathrm{False} \mid C_1 \wedge C_2 \mid \exists a_?.C \mid A \stackrel{?}{=} B$

And a *constraint generation* function ${}_\Gamma[\![t]\!]_A$ such that

$$\begin{array}{rcl} \gamma \vdash {}_\Gamma[\![t]\!]_A & \implies & \gamma(\Gamma) \vdash t : \gamma(A) \\ \Gamma \vdash t : A & \implies & \exists \gamma.\, \gamma \vdash {}_\Gamma[\![t]\!]_A \end{array}$$

$$\gamma \vdash \mathrm{True} \qquad \frac{\gamma \vdash C_1 \quad \gamma \vdash C_2}{\gamma \vdash C_1 \wedge C_2} \qquad \frac{\gamma[a_? := A] \vdash C}{\gamma \vdash \exists a_?.\, C} \qquad \frac{\gamma(A) = \gamma(B)}{\gamma \vdash A \stackrel{?}{=} B}$$

Questions:

- is this enough?

Meta-theory

Define a language of *constraints*: $C ::= \mathrm{True} \mid \mathrm{False} \mid C_1 \wedge C_2 \mid \exists a_?.C \mid A \stackrel{?}{=} B$

And a *constraint generation* function ${}_{\Gamma}[\![t]\!]_A$ such that

$$\begin{array}{lcl} \gamma \vdash {}_{\Gamma}[\![t]\!]_A & \Longrightarrow & \gamma(\Gamma) \vdash t : \gamma(A) \\ \Gamma \vdash t : A & \Longrightarrow & \exists \gamma.\, \gamma \vdash {}_{\Gamma}[\![t]\!]_A \end{array}$$

$$\gamma \vdash \mathrm{True} \qquad \frac{\gamma \vdash C_1 \quad \gamma \vdash C_2}{\gamma \vdash C_1 \wedge C_2} \qquad \frac{\gamma[a_? := A] \vdash C}{\gamma \vdash \exists a_?.\, C} \qquad \frac{\gamma(A) = \gamma(B)}{\gamma \vdash A \stackrel{?}{=} B}$$

Questions:

- is this enough? (several solutions?)

Cool “applicative” approach: [Pottier 2014]

“let”-polymorphism

Idea:

$$\begin{aligned} \emptyset \vdash \lambda x.\, \lambda y.\, x\ y : ? \qquad &\Longrightarrow \qquad a_? \to b_? \to c_? \text{ with } a_? = b_? \to c_? \\ &\Longrightarrow \qquad (b_? \to c_?) \to b_? \to c_? \\ &\Longrightarrow \qquad \forall \beta\gamma.(\beta \to \gamma) \to \beta \to \gamma \end{aligned}$$

Realisation:

$$\frac{\Gamma \vdash_{\text{inf}} t : A[\overline{a_?}] \qquad \overline{a_?} \notin \Gamma \qquad \Gamma, x{:}\forall\overline{\alpha}.A[\overline{a_?} := \overline{\alpha}] \vdash u : B}{\Gamma \vdash_{\text{inf}} \text{let } x = t \text{ in } u : B}$$

```
let f x =
  let y z = (x, z) in
  ...
```

Implementation

Unification constraints: **union-find** data structure.

(Live-coding exercise!)

Efficient generalization: variable **ranks** (or **levels**).

Whacky project ideas from the audience

- NbE for a dependently typed language
- Effect handlers
- Linear types / session types
- lambda-lambdabar-mu-mutilda (recursion and corecursion?)
- modal type systems / type-systems for stream programming

References

François Pottier. 2014. Hindley-Milner elaboration in applicative style. *ICFP*. http://cambium.inria.fr/~fpottier/publis/fpottier-elaboration.pdf.

Jana Dunfield and Neel Krishnaswami. 2021. Bidirectional Typing. *ACM Computing Surveys 54*, 5, 1–38. https://arxiv.org/abs/1908.05839.