Type inference

Niki Vazou, Summer 2015

Type inference

$40 + 2 : \mathrm{Int}$

Typing Rules

$$\frac{40{:}\mathrm{Int} \quad 2{:}\mathrm{Int}}{40+2{:}\mathrm{Int}}$$

If the premises (above line) hold,
then the conclusion (below line) holds.

Typing Rules

with variables

$$\frac{\Gamma \vdash x : \mathrm{Int} \quad \Gamma \vdash 2 : \mathrm{Int}}{\Gamma \vdash x + 2 : \mathrm{Int}}$$

Keep variable info in an *environment* Γ.

Here $\Gamma = \{x : \mathrm{Int}\}$

Typing Rules

the plus rule

$$\frac{\Gamma \vdash e1 : Int \quad \Gamma \vdash e2 : Int}{\Gamma \vdash e1 + e2 : Int}$$

Polymorphism

```
id x = x
id :: t -> t
```

type variable t stands for *any* type

Polymorphism

```
fst (x,y) = x
fst :: (t, t1) -> t
```

type variables t, t1 stand for *any* type

Polymorphism

```
strange f = (f 5, f “foo”)
type error
```

f is not polymorphic!

Type inference

```
let f g x   = g x (x + 1)
let m a b = a * b
f m 6
```

find the types

Type inference

```
let f (g::@1) (x::@2) :: @3 = g x (x + 1)
let m (a::@4) (b::@5) :: @6 = a * b
f m 6
```

add type variables for all the unknown types

```
f :: @1 -> @2 -> @3,
m :: @4 -> @5 -> @6
```

Type inference

```
let f (g::@1) (x::@2) :: @3 = g x (x + 1)
let m (a::@4) (b::@5) :: @6 = a * b
f m 6
```

add constraints

```
f :: @1 -> @2 -> @3,
m :: @4 -> @5 -> @6,
@4 := Int, @5 := Int, @6 := Int
```

Type inference

```
let f (g::@1) (x::@2) :: @3 = g x (x + 1)
let m (a::@4) (b::@5) :: @6 = a * b
f m 6
```

add constraints

```
f :: @1 -> @2 -> @3,
m :: @4 -> @5 -> @6,
@4 := Int, @5 := Int, @6 := Int,
@2 := Int, @1 := Int -> Int -> @3
```

Type inference

```
let f (g::@1) (x::@2) :: @3 = g x (x + 1)
let m (a::@4) (b::@5) :: @6 = a * b
f m 6
```

add constraints

```
f :: @1 -> @2 -> @3,
m :: @4 -> @5 -> @6,
@4 := Int, @5 := Int, @6 := Int,
@2 := Int, @1 := Int -> Int -> @3,
@1 := Int -> Int -> Int
```

Type inference

```
f :: @1 -> @2 -> @3,
m :: @4 -> @5 -> @6,
@4 := Int, @5 := Int, @6 := Int,
@2 := Int, @1 := Int -> Int -> @3,
@1 := Int -> Int -> Int
```

Type inference in λ-calculus

expressions: $e ::= x \mid \lambda x.e \mid e\ e$

types: $t ::= \alpha \mid t \rightarrow t$

$$\frac{\Gamma \vdash e : t_x \rightarrow t \quad \Gamma \vdash e_x : t_x}{\Gamma \vdash e\ e_x : t}$$

$$\frac{(x:t) \in \Gamma}{\Gamma \vdash x : t}$$

$$\frac{\Gamma, x:t_x \vdash e : t}{\Gamma \vdash \lambda x.e : t_x \rightarrow t}$$

Type inference in λ-calculus

constraint solving

unify(∅) = {}
unify({ τ1 = τ2 } ∪ C) =
  if τ1 ≡ τ2 then
    unify(C)
  else if τ1 ≡ α and α not in τ2 then
    unify([α → τ2] ∪ C) ○ [α → τ2] — compose unifications
  else if τ2 ≡ α and α not in τ1 then
    unify([α → τ1]C) ○ [α → τ1]
  else if τ1 ≡ τ11 → τ12 and τ2 ≡ τ21 → τ22 then
    unify(C ∪ { τ11 = τ21, τ12 = τ22 })
  else
    failure!