HM(X): Type Inference with Constraint Types

Marius Weidner

Chair of Programming Languages, University of Freiburg

July 10, 2023

Outline

Hindley Milner Basics

- Let Polymorphism
- Full Type Inference
- Principle Type Property

Pseudocode Example

```
let id = λx. x in             :: α → α
id 42                         :: Int
id "Hello World"              :: Str
```

Hindley Milner Syntax

$$
\begin{aligned}
e \;::=\;& x \\
\mid\;& \lambda\mathsf{x}.\ \mathsf{e} \\
\mid\;& \mathsf{e}\ \ \mathsf{e} \\
\mid\;& \textbf{let}\ \mathsf{x} = \mathsf{e}\ \ \textbf{in}\ \mathsf{e}
\end{aligned}
$$

$$\tau \;::=\; \alpha \mid \tau \rightarrow \tau$$

$$\sigma \;::=\; \tau \mid \forall \alpha . \sigma$$

Hindley Milner Typing

$$\frac{\Gamma \vdash e : \sigma \quad \Gamma, x : \sigma \vdash e' : \tau'}{\Gamma \vdash \textbf{let } x = e \textbf{ in } e' : \tau'} \text{ Let}$$

$$\frac{\Gamma \vdash e : \sigma \quad \text{fresh } \alpha}{\Gamma \vdash e : \forall \alpha . \sigma} \ \forall \text{ Intro}$$

$$\frac{\Gamma \vdash e : \sigma' \quad \sigma' \sqsubseteq \sigma}{\Gamma \vdash e : \sigma} \ \forall \text{ Elim}$$

HM(X) Syntax

$$
\begin{aligned}
C \ &::= \ x \\
&\ \mid \ \lambda\text{x. e} \\
&\ \mid \ \text{e e} \\
&\ \mid \ \textbf{let}\ \text{x} = \text{e}\ \textbf{in}\ \text{e}
\end{aligned}
$$

$$\tau \ ::= \ \alpha \mid \tau \to \tau$$

$$\sigma \ ::= \ \tau \mid \forall \alpha . C \Rightarrow \sigma$$

HM(X) Typing

$$\frac{C, \Gamma \vdash e : \sigma \quad C, (\Gamma, x : \sigma) \vdash e' : \tau'}{C, \Gamma \vdash \textbf{let } x = e \textbf{ in } e' : \tau'} \text{ Let}$$

$$\frac{C \wedge D, \Gamma \vdash e : \tau \quad \text{fresh } \vec{\alpha}}{C \wedge \exists \vec{\alpha}.D, \Gamma \vdash e : \forall \vec{\alpha}.D \Rightarrow \tau} \ \forall \text{ Intro}$$

$$\frac{C, \Gamma \vdash e : \forall \vec{\alpha}.D \Rightarrow \tau' \quad C \vdash^e [\vec{\tau}/\vec{\alpha}]D}{C, \Gamma \vdash e : [\vec{\tau}/\vec{\alpha}]\tau'} \ \forall \text{ Elim}$$

.

.