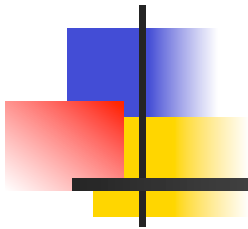


Programming Languages and Compilers (CS 421)

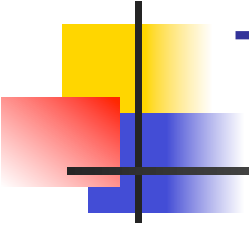


Elsa L Gunter

2112 SC, UIUC

<http://courses.engr.illinois.edu/cs421>

Based in part on slides by Mattox Beckman, as updated by Vikram Adve and Gul Agha



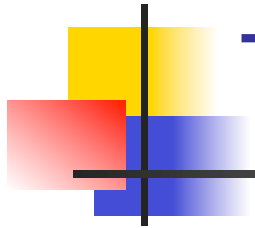
Two Problems

- Type checking

- Question: Does exp. e have type τ in env Γ ?
- Answer: Yes / No
- Method: Type **derivation**

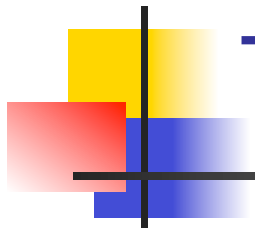
- Typability

- Question Does exp. e have **some type** in env. Γ ?
If so, what is it?
- Answer: Type τ / error
- Method: Type **inference**



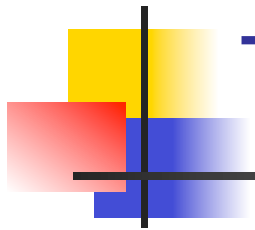
Type Inference - Outline

- Begin by assigning a type variable as the type of the whole expression
- Decompose the expression into component expressions
- Use typing rules to generate constraints on components and whole
- Recursively find substitution that solves typing judgment of first subcomponent
- Apply substitution to next subcomponent and find substitution solving it; compose with first, etc.
- Apply comp of all substitution to orig. type var. to get answer



Type Inference - Example

- What type can we give to
 $(\text{fun } x \rightarrow \text{fun } f \rightarrow f \ x)$
- Start with a type variable and then look at the way the term is constructed



Type Inference - Example

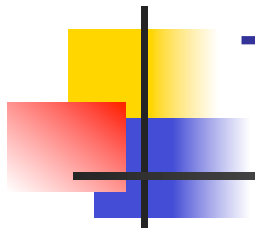
- First approximate:

$$[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha$$

- Second approximate: use fun rule

$$\frac{[x : \beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}{[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha}$$

- Remember constraint $\alpha \equiv (\beta \rightarrow \gamma)$



Type Inference - Example

- Third approximate: use fun rule

$$\frac{\frac{[f : \delta ; x : \beta] \vdash f (f x) : \varepsilon}{[x : \beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}}{[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f(f x)) : \alpha}$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

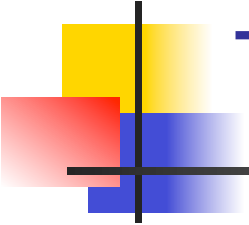
- Fourth approximate: use app rule

$$\frac{[f:\delta; x:\beta] \vdash f : \varphi \rightarrow \varepsilon \quad [f:\delta; x:\beta] \vdash f x : \varphi}{[f : \delta ; x : \beta] \vdash (f (f x)) : \varepsilon}$$

$$\frac{[x : \beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}{[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha}$$

$$[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$



Type Inference - Example

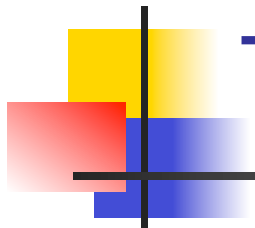
- Fifth approximate: use var rule, get constraint $\delta \equiv \varphi \rightarrow \varepsilon$, Solve with same
- Apply to next sub-proof

$$\frac{[f:\delta; x:\beta] \vdash f : \varphi \rightarrow \varepsilon \quad [f:\delta; x:\beta] \vdash f x : \varphi}{[f:\delta; x:\beta] \vdash f (f x) : \varepsilon}$$

$$\frac{[f:\delta; x:\beta] \vdash f (f x) : \varepsilon}{[x:\beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}$$

$$\frac{[x:\beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}{[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha}$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$



Type Inference - Example

- Current subst: $\{\delta \equiv \varphi \rightarrow \varepsilon\}$

$$\frac{\dots \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f \ x : \varphi}{[f : \delta ; x : \beta] \vdash (f (f \ x)) : \varepsilon}$$
$$\frac{[x : \beta] \vdash (\text{fun } f \rightarrow f (f \ x)) : \gamma}{[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f \ x)) : \alpha}$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

- Current subst: $\{\delta \equiv \varphi \rightarrow \varepsilon\}$

$$\begin{array}{c}
 [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f:\zeta \rightarrow \varphi \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash x:\zeta \\
 \hline
 \dots \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f\ x : \varphi \\
 \hline
 [f : \delta ; x : \beta] \vdash (f (f\ x)) : \varepsilon \\
 \hline
 [x : \beta] \vdash (\text{fun } f \text{ -> } f (f\ x)) : \gamma \\
 \hline
 [] \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f (f\ x)) : \alpha
 \end{array}$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

- Current subst: $\{\delta \equiv \varphi \rightarrow \varepsilon\}$
- Var rule: Solve $\zeta \rightarrow \varphi \equiv \varphi \rightarrow \varepsilon$ **Unification**

$$[f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f:\zeta \rightarrow \varphi \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash x:\zeta$$

$$\dots \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f\ x : \varphi$$

$$[f : \delta ; x : \beta] \vdash (f (f\ x)) : \varepsilon$$

$$[x : \beta] \vdash (\text{fun } f \text{ -> } f (f\ x)) : \gamma$$

$$[] \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f (f\ x)) : \alpha$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

- Current subst: $\{\zeta \equiv \varepsilon, \varphi \equiv \varepsilon\} \circ \{\delta \equiv \varphi \rightarrow \varepsilon\}$
- Var rule: Solve $\zeta \rightarrow \varphi \equiv \varphi \rightarrow \varepsilon$ **Unification**

$$\frac{[f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f:\zeta \rightarrow \varphi \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash x:\zeta}{\dots}$$

$$\frac{\dots \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f \ x : \varphi}{[f : \delta ; x : \beta] \vdash (f (f \ x)) : \varepsilon}$$

$$\frac{[x : \beta] \vdash (\text{fun } f \text{ -> } f (f \ x)) : \gamma}{[] \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f (f \ x)) : \alpha}$$

$$[] \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f (f \ x)) : \alpha$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

- Current subst: $\{\zeta \equiv \varepsilon, \varphi \equiv \varepsilon, \delta \equiv \varepsilon \rightarrow \varepsilon\}$
- Apply to next sub-proof

$$\begin{array}{c}
 \dots \quad [f:\varepsilon \rightarrow \varepsilon; x:\beta] \vdash x:\varepsilon \\
 \hline
 \dots \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f\ x : \varphi \\
 \hline
 [f : \delta ; x : \beta] \vdash (f (f\ x)) : \varepsilon \\
 \hline
 [x : \beta] \vdash (\text{fun } f \text{ -> } f (f\ x)) : \gamma \\
 \hline
 [] \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f (f\ x)) : \alpha
 \end{array}$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

- Current subst: $\{\zeta \equiv \varepsilon, \varphi \equiv \varepsilon, \delta \equiv \varepsilon \rightarrow \varepsilon\}$
- Var rule: $\varepsilon \equiv \beta$

$$\begin{array}{c}
 \dots \quad \frac{}{[f:\varepsilon \rightarrow \varepsilon; x:\beta] \vdash x:\varepsilon} \\
 \hline
 \dots \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f\ x : \varphi \\
 \hline
 [f : \delta ; x : \beta] \vdash (f (f\ x)) : \varepsilon \\
 \hline
 [x : \beta] \vdash (\text{fun } f \text{ -> } f (f\ x)) : \gamma \\
 \hline
 [] \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f (f\ x)) : \alpha
 \end{array}$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

- Current subst: $\{\varepsilon \equiv \beta\} \circ \{\zeta \equiv \varepsilon, \varphi \equiv \varepsilon, \delta \equiv \varepsilon \rightarrow \varepsilon\}$
- Solves subproof; return one layer

$$\begin{array}{c}
 \dots \quad \dfrac{}{[f:\varepsilon \rightarrow \varepsilon; x:\beta] \vdash x:\varepsilon} \\
 \hline
 \dots \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f\ x : \varphi \\
 \hline
 [f : \delta ; x : \beta] \vdash (f\ (f\ x)) : \varepsilon \\
 \hline
 [x : \beta] \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma \\
 \hline
 [] \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha
 \end{array}$$

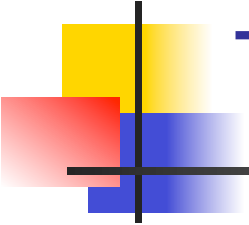
- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

- Current subst: $\{\varepsilon \equiv \beta, \zeta \equiv \beta, \varphi \equiv \beta, \delta \equiv \beta \rightarrow \beta\}$
- Solves this subproof; return one layer

$$\begin{array}{c}
 \hline
 \dots \quad [f:\varphi \rightarrow \varepsilon; x:\beta] \vdash f\ x : \varphi \\
 \hline
 [f : \delta ; x : \beta] \vdash (f\ (f\ x)) : \varepsilon \\
 \hline
 [x : \beta] \vdash (\text{fun } f \rightarrow f\ (f\ x)) : \gamma \\
 \hline
 [] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f\ (f\ x)) : \alpha
 \end{array}$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$



Type Inference - Example

- Current subst: $\{\varepsilon \equiv \beta, \zeta \equiv \beta, \varphi \equiv \beta, \delta \equiv \beta \rightarrow \beta\}$
- Need to satisfy constraint $\gamma \equiv (\delta \rightarrow \varepsilon)$,
given subst: $\gamma \equiv ((\beta \rightarrow \beta) \rightarrow \beta)$

$$\frac{\frac{[f : \delta ; x : \beta] \vdash (f (f x)) : \varepsilon}{[x : \beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}}{[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha}$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

- Current subst:

$$\{\gamma \equiv ((\beta \rightarrow \beta) \rightarrow \beta), \varepsilon \equiv \beta, \zeta \equiv \beta, \varphi \equiv \beta, \delta \equiv \beta \rightarrow \beta\}$$

- Solves subproof; return one layer

$$\frac{}{[f : \delta ; x : \beta] \vdash (f (f x)) : \varepsilon}$$

$$\frac{}{[x : \beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}$$

$$[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha$$

- $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon)$

Type Inference - Example

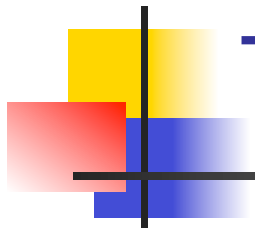
- Current subst:

$$\{\gamma \equiv ((\beta \rightarrow \beta) \rightarrow \beta), \varepsilon \equiv \beta, \zeta \equiv \beta, \varphi \equiv \beta, \delta \equiv \beta \rightarrow \beta\}$$

- Need to satisfy constraint $\alpha \equiv (\beta \rightarrow \gamma)$
given subst: $\alpha \equiv (\beta \rightarrow ((\beta \rightarrow \beta) \rightarrow \beta))$

$$\frac{[x : \beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}{[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha}$$

- $\alpha \equiv (\beta \rightarrow \gamma);$



Type Inference - Example

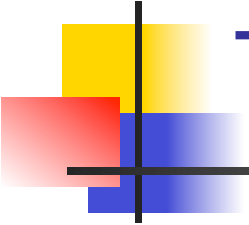
- Current subst:

$\{\alpha \equiv (\beta \rightarrow ((\beta \rightarrow \beta) \rightarrow \beta)),$

$\gamma \equiv ((\beta \rightarrow \beta) \rightarrow \beta), \varepsilon \equiv \beta, \zeta \equiv \beta, \varphi \equiv \beta, \delta \equiv \beta \rightarrow \beta\}$

- Solves subproof; return on layer

$$\frac{[x : \beta] \vdash (\text{fun } f \rightarrow f (f x)) : \gamma}{[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha}$$



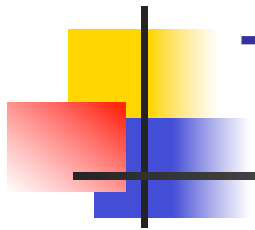
Type Inference - Example

- Current subst:

$\{\alpha \equiv (\beta \rightarrow ((\beta \rightarrow \beta) \rightarrow \beta)),$
 $\gamma \equiv ((\beta \rightarrow \beta) \rightarrow \beta), \varepsilon \equiv \beta, \zeta \equiv \beta, \varphi \equiv \beta, \delta \equiv \beta \rightarrow \beta\}$

- Done: $\alpha \equiv (\beta \rightarrow ((\beta \rightarrow \beta) \rightarrow \beta))$

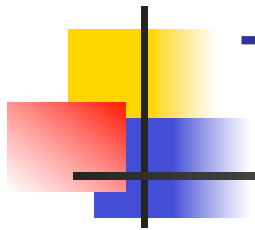
$[] \vdash (\text{fun } x \rightarrow \text{fun } f \rightarrow f (f x)) : \alpha$



Type Inference Algorithm

Let $\text{infer}(\Gamma, e, \tau) = \sigma$

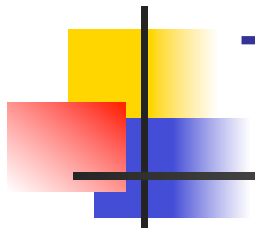
- Γ is a typing environment (giving polymorphic types to expression variables)
- e is an expression
- τ is a type (with type variables),
- σ is a substitution of types for type variables
- Idea: σ is the constraints on type variables necessary for $\Gamma \vdash e : \tau$
- Should have $\sigma(\Gamma) \vdash e : \sigma(\tau)$



Type Inference Algorithm

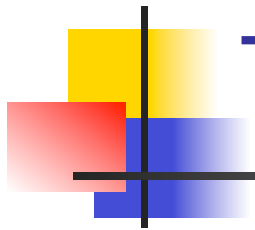
has_type (Γ, exp, τ) =

- Case *exp* of
 - Var $v \rightarrow$ return $\text{Unify}\{\tau \equiv \text{freshInstance}(\Gamma(v))\}$
 - Replace all quantified type vars by fresh ones
 - Const $c \rightarrow$ return $\text{Unify}\{\tau \equiv \text{freshInstance } \varphi \}$
where $\Gamma \vdash c : \varphi$ by the constant rules
 - fun $x \rightarrow e \rightarrow$
 - Let α, β be fresh variables
 - Let $\sigma = \text{infer}([x: \alpha] + \Gamma, e, \beta)$
 - Return $\text{Unify}(\{\sigma(\tau) \equiv \sigma(\alpha \rightarrow \beta)\}) \circ \sigma$



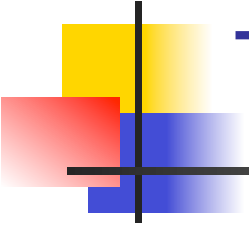
Type Inference Algorithm (cont)

- Case *exp* of
 - App (e_1 e_2) $\rightarrow$
 - Let α be a fresh variable
 - Let $\sigma_1 = \text{infer}(\Gamma, e_1, \alpha \rightarrow \tau)$
 - Let $\sigma_2 = \text{infer}(\sigma(\Gamma), e_2, \sigma(\alpha))$
 - Return $\sigma_2 \circ \sigma_1$



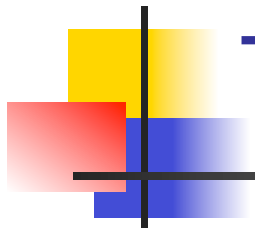
Type Inference Algorithm (cont)

- Case *exp* of
 - If e_1 then e_2 else $e_3 \rightarrow$
 - Let $\sigma_1 = \text{infer}(\Gamma, e_1, \text{bool})$
 - Let $\sigma_2 = \text{infer}(\sigma\Gamma, e_2, \sigma_1(\tau))$
 - Let $\sigma_3 = \text{infer}(\sigma_2 \circ \sigma_1(\Gamma), e_3, \sigma_2 \circ \sigma(\tau))$
 - Return $\sigma_3 \circ \sigma_2 \circ \sigma_1$



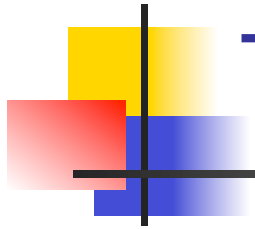
Type Inference Algorithm (cont)

- Case *exp* of
 - let $x = e_1$ in $e_2 \rightarrow$
 - Let α be a fresh variable
 - Let $\sigma_1 = \text{infer}(\Gamma, e_1, \alpha)$
 - Let $\sigma_2 =$
 $\text{infer}([x:\text{GEN}(\sigma_1(\Gamma), \sigma_1(\alpha))] + \sigma_1(\Gamma),$
 $e_2, \sigma_1(\tau))$
 - Return $\sigma_2 \circ \sigma_1$



Type Inference Algorithm (cont)

- Case *exp* of
 - let rec $x = e_1$ in $e_2 \rightarrow$
 - Let α be a fresh variable
 - Let $\sigma_1 = \text{infer}([x: \alpha] + \Gamma, e_1, \alpha)$
 - Let $\sigma_2 =$
 $\text{infer}([x: [x: \text{GEN}(\sigma_1(\Gamma), \sigma_1(\alpha))] + \sigma_1(\Gamma)]$
 $+ \sigma_1(\Gamma), e_2, \sigma_1(\tau))$
 - Return $\sigma_2 \circ \sigma_1$



Type Inference Algorithm (cont)

- To infer a type, introduce `type_of`
- Let α be a fresh variable
- `type_of` (Γ, e) =
 - Let $\sigma = \text{infer}(\Gamma, e, \alpha)$
 - Return $\sigma(\alpha)$
- Need an algorithm for Unif