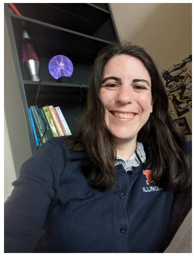
Programming Languages and Compilers (CS 421)

Talia Ringer (they/them)

4218 SC, UIUC



https://courses.grainger.illinois.edu/cs421/fa2023/

Based heavily on slides by Elsa Gunter, which were based in part on slides by Mattox Beckman, as updated by Vikram Adve and Gul Agha

Objectives for Today

- Last week, we covered **type checking** as it interacted with **polymorphism**
- This week, we will continue to **type inference**

Questions from last week?

Type Inference

Two Problems

- Type checking
  - Question: Does exp. e have type T 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 T / error
  - Method: Type inference

Two Problems

- Type checking
  - Question: Does exp. e have type T 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 T / error
  - Method: Type inference

Type Inference - Outline

- **Assign a type variable** to the whole expression
- **Decompose the expression** into components
- **Generate constraints** using typing rules, both on components and on the whole expression
- **Recursively find a substitution** that solves the typing judgment of the first component
- **Apply substitution** to the next component; **find substitution** solving the next component; **compose** with the first, etc.
- **Apply composition of all substitutions** to the original type variable to get the answer

Type Inference - Outline

- **Assign a type variable** to the whole expression
- **Decompose the expression** into components
- **Generate constraints** using typing rules, both on components and on the whole expression
- **Recursively find a substitution** that solves the typing judgment of the first component
- **Apply substitution** to the next component; **find substitution** solving the next component; **compose** with the first, etc.
- **Apply composition of all substitutions** to the original type variable to get the answer

Type Inference - Outline

- **Assign a type variable** to the whole expression
- **Decompose the expression** into components
- **Generate constraints** using typing rules, both on components and on the whole expression
- **Recursively find a substitution** that solves the typing judgment of the first component
- **Apply substitution** to the next component; **find substitution** solving the next component; **compose** with the first, etc.
- **Apply composition of all substitutions** to the original type variable to get the answer

Type Inference - Outline

- **Assign a type variable** to the whole expression
- **Decompose the expression** into components
- **Generate constraints** using typing rules, both on components and on the whole expression
- **Recursively find a substitution** that solves the typing judgment of the first component
- **Apply substitution** to the next component; **find substitution** solving the next component; **compose** with the first, etc.
- **Apply composition of all substitutions** to the original type variable to get the answer

Type Inference - Outline

- **Assign a type variable** to the whole expression
- **Decompose the expression** into components
- **Generate constraints** using typing rules, both on components and on the whole expression
- **Recursively find a substitution** that solves the typing judgment of the first component
- **Apply substitution** to the next component; **find substitution** solving the next component; **compose** with the first, etc.
- **Apply composition of all substitutions** to the original type variable to get the answer

Type Inference - Outline

- **Assign a type variable** to the whole expression
- **Decompose the expression** into components
- **Generate constraints** using typing rules, both on components and on the whole expression
- **Recursively find a substitution** that solves the typing judgment of the first component
- **Apply substitution** to the next component; **find substitution** solving the next component; **compose** with the first, etc.
- **Apply composition of all substitutions** to the original type variable to get the answer

Type Inference - Outline

- **Assign a type variable** to the whole expression
- **Decompose the expression** into components
- **Generate constraints** using typing rules, both on components and on the whole expression
- **Recursively find a substitution** that solves the typing judgment of the first component
- **Apply substitution** to the next component; **find substitution** solving the next component; **compose** with the first, etc.
- **Apply composition of all substitutions** to the original type variable to get the answer

Type Inference - Outline

- **Assign a type variable** to the whole expression
- **Decompose the expression** into components
- **Generate constraints** using typing rules, both on components and on the whole expression
- **Recursively find a substitution** that solves the typing judgment of the first component
- **Apply substitution** to the next component; **find substitution** solving the next component; **compose** with the first, etc.
- **Apply composition of all substitutions** to the original type variable to get the answer

Questions so far?

Type Inference Example

Type Inference - Example

- What type can we give to:

  (fun x -> fun f -> f (f x))
- **Assign a type variable** and then look at the way the term is constructed
- Trying to find an α such that:

  $\{ \} \vdash$ (fun x -> fun f -> f (f x)) : α

$$\frac{???}{\{ \} \vdash \text{(fun x -> fun f -> f (f x))} : \alpha}$$

Example

Type Inference - Example

- What type can we give to:

  (fun x -> fun f -> f (f x))

- **Assign a type variable** and then look at the way the term is constructed
- Trying to find an α such that:

  { } ⊢ (fun x -> fun f -> f (f x)) : α

$$\frac{???}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}$$

Example

Type Inference - Example

- What type can we give to:

  (fun x -> fun f -> f (f x))
- **Assign a type variable** and then look at the way the term is constructed
- Trying to find an α such that:

  { } ⊢ (fun x -> fun f -> f (f x)) : α

$$\frac{???}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}$$

Type Inference - Example

- What type can we give to:

  (fun x -> fun f -> f (f x))

- **Assign a type variable** and then look at the way the term is constructed
- Trying to find an α such that:

  { } ⊢ (fun x -> fun f -> f (f x)) : α

$$\frac{\textbf{???}}{\{\ \} \vdash (\text{fun x -> fun f -> f (f x)}) : \alpha}$$

Example

Type Inference - Example

$$\frac{???}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}$$

Example

Type Inference - Example

Decompose the expression into components

$$\frac{???}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}$$

Example

Type Inference - Example

Function rule

Decompose the expression into components

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

Example

Type Inference - Example

Function rule

Generate constraints
using the typing rule

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

Example

Type Inference - Example

Function rule

Generate constraints
using the typing rule

$$\frac{\{x : \boldsymbol{\beta}\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \boldsymbol{\gamma}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \boldsymbol{\alpha}}\ \textbf{FUN}$$

Constraints: $\boldsymbol{\alpha} \equiv (\boldsymbol{\beta} \rightarrow \boldsymbol{\gamma})$

Example

Type Inference - Example

Recursively find substitution

$$\frac{\frac{???}{\{x : \beta\} \vdash (\text{fun f -> f (f x)}) : \gamma}}{\{\,\} \vdash (\text{fun x -> fun f -> f (f x)}) : \alpha} \;\textbf{FUN}$$

Constraints: $\alpha \equiv (\beta \rightarrow \gamma)$

Example

Type Inference - Example

Decompose the expression into components

???

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

Constraints: $\alpha \equiv (\beta \rightarrow \gamma)$

Example

Type Inference - Example

Function rule

Decompose the expression into components

$$\frac{\dfrac{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\{x : \beta\} \vdash (\text{fun } f \to f\ (f\ x)) : \gamma}\ \textbf{FUN}}{\{\ \} \vdash (\text{fun } x \to \text{fun } f \to f\ (f\ x)) : \alpha}\ \textbf{FUN}$$

Constraints: $\alpha \equiv (\beta \rightarrow \gamma)$

Example

Type Inference - Example

Function rule

Generate constraints using the typing rule

$$\dfrac{\dfrac{\{f : \delta \,;\, x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\{x : \beta\} \vdash (\text{fun } f \to f\ (f\ x)) : \gamma}\ \textbf{FUN}}{\{\ \} \vdash (\text{fun } x \to \text{fun } f \to f\ (f\ x)) : \alpha}\ \textbf{FUN}$$

Constraints: $\alpha \equiv (\beta \to \gamma)$

Example

Type Inference - Example

Function rule

Generate constraints
using the typing rule

$$\cfrac{\cfrac{\{f : \boldsymbol{\delta} \,;\, x : \beta\} \vdash f\ (f\ x) : \boldsymbol{\varepsilon}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \boldsymbol{\gamma}}\ \textbf{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\ \textbf{FUN}$$

Constraints: $\alpha \equiv (\beta \to \gamma)$; $\boldsymbol{\gamma} \equiv (\boldsymbol{\delta} \to \boldsymbol{\varepsilon})$

Example

Type Inference - Example

Recursively find substitution

???

$$\frac{\dfrac{\dfrac{???}{\{f : \delta \,;\, x : \beta\} \vdash f\ (f\ x) : \varepsilon}\ \text{FUN}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\ \text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}$$

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

Example

Questions so far?

Example

Type Inference - Example

Recursively find substitution

$$\dfrac{\dfrac{\dfrac{???}{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}\ \text{FUN}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\ \text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}$$

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

Type Inference - Example

Decompose the expression into components

???

$$\frac{\dfrac{\dfrac{???}{\{f : \delta \,;\, x : \beta\} \vdash f\ (f\ x) : \varepsilon}\ \text{FUN}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\ \text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}$$

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

Example

Type Inference - Example

Application rule

Decompose the expression into components

$$\dfrac{\dfrac{\dfrac{\{f : \delta\ ;\ x : \beta\} \vdash f : \phi \rightarrow \varepsilon \qquad \{f : \delta\ ;\ x : \beta\} \vdash f\ x : \phi}{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}\ \text{APP}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\ \text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\ \text{FUN}$$

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

Type Inference - Example

Nothing is known about ϕ yet, but we'll learn more soon.

$$\dfrac{\dfrac{\dfrac{\{f : \delta\ ;\ x : \beta\} \vdash f : \boldsymbol{\phi} \rightarrow \varepsilon \quad \{f : \delta\ ;\ x : \beta\} \vdash f\ x : \boldsymbol{\phi}}{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}\ \text{APP}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\ \text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\ \text{FUN}$$

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

Example

Type Inference - Example

Recursively find substitution

$$\dfrac{\dfrac{\color{red}{???}}{\{f : \delta\ ;\ x : \beta\} \vdash f : \phi \to \varepsilon} \quad \{f : \delta\ ;\ x : \beta\} \vdash f\ x : \phi}{\dfrac{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> } \text{fun } f \text{ -> } f\ (f\ x)) : \alpha}\ \text{FUN}}\ \text{FUN}}\ \text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Example

Type Inference - Example

Decompose the expression into components

???

$$\{f : \delta ; x : \beta\} \vdash f : \phi \to \varepsilon \quad \{f : \delta ; x : \beta\} \vdash f\ x : \phi$$ APP

$$\{f : \delta ; x : \beta\} \vdash f\ (f\ x) : \varepsilon$$ FUN

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

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

Constraints: $\alpha \equiv (\beta \to \gamma); \gamma \equiv (\delta \to \varepsilon)$

Example

Type Inference - Example

Decompose the expression into components

Var rule, since f is bound in environment {f : δ ; x : β}.

$$\dfrac{\dfrac{}{\{f : \delta ; x : \beta\} \vdash f : \phi \rightarrow \varepsilon}\text{VAR} \quad \{f : \delta ; x : \beta\} \vdash f\ x : \phi}{\dfrac{\{f : \delta ; x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

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

Type Inference - Example

Generate constraints using the typing rule

Var rule, since f is bound in environment {f : δ ; x : β}.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{f : \delta\ ;\ x : \beta\} \vdash f : \phi \rightarrow \varepsilon}\ \text{VAR} \qquad \{f : \delta\ ;\ x : \beta\} \vdash f\ x : \phi}{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}\ \text{APP}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\ \text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> } \text{fun } f \text{ -> } f\ (f\ x)) : \alpha}\ \text{FUN}$$

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

Example

Type Inference - Example

Generate constraints
using the typing rule

Var rule, since f is bound in environment {f : δ ; x : β}.

$$\dfrac{\dfrac{}{\{f : \boldsymbol{\delta} ; x : \beta\} \vdash f : \boldsymbol{\phi \to \varepsilon}}\text{VAR} \quad \{f : \delta ; x : \beta\} \vdash f\ x : \phi}{\dfrac{\{f : \delta ; x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma)$; $\gamma \equiv (\delta \to \varepsilon)$; $\boldsymbol{\delta \equiv (\phi \to \varepsilon)}$

Example

Type Inference - Example

Apply substitution
(since the LHS is done)

$$\dfrac{\dfrac{\dfrac{\overline{\{f : \delta ; x : \beta\} \vdash f : \phi \rightarrow \varepsilon}^{\text{VAR}} \quad \{f : \delta ; x : \beta\} \vdash f\ x : \phi}{\{f : \delta ; x : \beta\} \vdash f\ (f\ x) : \varepsilon}\text{APP}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}$$

Constraints: $\alpha \equiv (\beta \rightarrow \gamma); \gamma \equiv (\delta \rightarrow \varepsilon); \delta \equiv (\phi \rightarrow \varepsilon)$

Example

Type Inference - Example

Apply substitution
(since the LHS is done)

VAR

$$\{f : \boldsymbol{\delta} ; x : \beta\} \vdash f : \phi \rightarrow \varepsilon \qquad \{f : \boldsymbol{\delta} ; x : \beta\} \vdash f\ x : \phi$$

APP

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

FUN

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

FUN

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

Constraints: $\alpha \equiv (\beta \rightarrow \gamma)$; $\gamma \equiv (\delta \rightarrow \varepsilon)$; $\boldsymbol{\delta \equiv (\phi \rightarrow \varepsilon)}$

Type Inference - Example

Apply substitution
(since the LHS is done)

$$\dfrac{\dfrac{\dfrac{\overline{\{f: \boldsymbol{\phi \to \varepsilon}; x:\beta\} \vdash f : \phi \to \varepsilon}^{\text{VAR}} \quad \{f: \boldsymbol{\phi \to \varepsilon}; x:\beta\} \vdash f\ x : \phi}{\{f : \delta\ ; x : \beta\} \vdash f\ (f\ x) : \varepsilon}\ \text{APP}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\ \text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\ \text{FUN}$$

Constraints: $\alpha \equiv (\beta \to \gamma); \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\boldsymbol{\{\delta \equiv \phi \to \varepsilon\}}$

Type Inference - Example

Recursively find substitution

???

$$\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\}\vdash f:\phi\to\varepsilon}\ \text{VAR}\qquad \dfrac{???}{\{f:\phi\to\varepsilon;\ x:\beta\}\vdash f\ x:\phi}}{\dfrac{\{f:\delta\ ;\ x:\beta\}\vdash f\ (f\ x):\varepsilon}{\dfrac{\{x:\beta\}\vdash (\text{fun } f \text{ -> } f\ (f\ x)):\gamma}{\{\ \}\vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)):\alpha}\ \text{FUN}}\ \text{FUN}}\ \text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Decompose the expression into components

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\ \text{VAR} \qquad \dfrac{\color{red}{???}}{\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}}{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}\ \text{APP}}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\ \text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\ \text{FUN}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Application rule

Decompose the expression into components

$$\{f:\phi\to\varepsilon;\ x:\beta\} \qquad \{f:\phi\to\varepsilon;\ x:\beta\}$$

$$\vdash f : \zeta\to\phi \qquad \vdash x : \zeta \quad \text{APP}$$

VAR

$$\{f:\ \phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon \qquad \{f:\ \phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi \quad \text{APP}$$

$$\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon \quad \text{FUN}$$

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

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

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Example

Type Inference - Example

Recursively find substitution

$$
\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\ \text{VAR} \qquad \dfrac{\dfrac{\text{???}}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \zeta\to\phi} \quad \{f:\phi\to\varepsilon;\ x:\beta\} \vdash x : \zeta}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}\ \text{APP}}{\dfrac{\{f:\delta\ ;\ x:\beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x:\beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\ \text{FUN}}\ \text{FUN}}\ \text{APP}
$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

*Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Variable rule

Decompose the expression into components

$$\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f:\phi\to\varepsilon}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f:\zeta\to\phi}\text{VAR} \quad \{f:\phi\to\varepsilon;\ x:\beta\} \vdash x:\zeta}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f\ x:\phi}\text{APP}}{\dfrac{\dfrac{\{f:\delta\ ;\ x:\beta\} \vdash f\ (f\ x):\varepsilon}{\{x:\beta\} \vdash (\text{fun } f \to f\ (f\ x)):\gamma}\text{FUN}}{\{\ \} \vdash (\text{fun } x \to \text{fun } f \to f\ (f\ x)):\alpha}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Variable rule

Generate constraints using the typing rule

$$\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\}\vdash f:\phi\to\varepsilon}\,\text{VAR} \qquad \dfrac{\dfrac{}{\{f:\boldsymbol{\phi}\to\boldsymbol{\varepsilon};\ x:\beta\}\vdash f:\boldsymbol{\zeta}\to\boldsymbol{\phi}}\,\text{VAR} \qquad \{f:\phi\to\varepsilon;\ x:\beta\}\vdash x:\zeta}{\{f:\phi\to\varepsilon;\ x:\beta\}\vdash f\ x:\phi}\,\text{APP}}{\dfrac{\{f:\delta\ ;\ x:\beta\}\vdash f\ (f\ x):\varepsilon}{\dfrac{\{x:\beta\}\vdash (\text{fun } f \text{ -> } f\ (f\ x)):\gamma}{\{\ \}\vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)):\alpha}\,\text{FUN}}\,\text{FUN}}\,\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon);\ \boldsymbol{\zeta\to\phi \equiv \phi\to\varepsilon}$

Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Variable rule

Generate constraints using the typing rule

VAR

$$\overline{\{f:\boldsymbol{\phi}\to\boldsymbol{\varepsilon};\ x:\beta\} \vdash f : \boldsymbol{\zeta\to\phi}} \qquad \{f:\phi\to\varepsilon;\ x:\beta\} \vdash x : \zeta$$

VAR / APP

$$\overline{\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon} \qquad \overline{\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}$$

APP

$$\overline{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}$$

FUN

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

FUN

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

Constraints: $\alpha \equiv (\beta \to \gamma)$; $\gamma \equiv (\delta \to \varepsilon)$; $\boldsymbol{\zeta\to\phi \equiv \phi\to\varepsilon}$

Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Example

Type Inference - Example

By **unification** (which we will learn about soon), we get that **$\zeta \equiv \varepsilon$** and **$\phi \equiv \varepsilon$**.

VAR

$\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \zeta\to\phi$ $\quad$ $\{f:\phi\to\varepsilon;\ x:\beta\} \vdash x : \zeta$ APP

VAR

$\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon$ $\quad$ $\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi$ APP

$\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon$ FUN

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

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

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon);$ **$\zeta \equiv \varepsilon;\ \phi \equiv \varepsilon$**

Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Apply substitution
(since the LHS is done)

$$\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \zeta\to\phi}\text{VAR} \quad \{f:\phi\to\varepsilon;\ x:\beta\} \vdash x : \zeta}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}\text{APP}}{\dfrac{\{f:\delta\ ;\ x:\beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x:\beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon);\ \zeta \equiv \varepsilon;\ \phi \equiv \varepsilon$

Substituted: $\{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Apply substitution
(since the LHS is done)

VAR

$\{f:\boldsymbol{\phi}\rightarrow\varepsilon;\ x:\beta\}$ $\vdash f : \boldsymbol{\zeta}\rightarrow\boldsymbol{\phi}$ $\{f:\boldsymbol{\phi}\rightarrow\varepsilon;\ x:\beta\}$ $\vdash x : \boldsymbol{\zeta}$ APP

VAR

$\{f:\ \phi\rightarrow\varepsilon;\ x:\beta\} \vdash f : \phi\rightarrow\varepsilon$ $\{f:\ \phi\rightarrow\varepsilon;\ x:\beta\} \vdash f\ x : \phi$ APP

$\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon$ FUN

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

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

Constraints: $\alpha \equiv (\beta \rightarrow \gamma)$; $\gamma \equiv (\delta \rightarrow \varepsilon)$; $\boldsymbol{\zeta \equiv \varepsilon}$; $\boldsymbol{\phi \equiv \varepsilon}$

Substituted: $\{\delta \equiv \phi \rightarrow \varepsilon\}$

Type Inference - Example

Apply substitution
(since the LHS is done)

$$\dfrac{\dfrac{}{\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\,\text{VAR} \qquad \dfrac{\dfrac{}{\{f:\boldsymbol{\varepsilon}\to\varepsilon;\ x:\beta\} \vdash f : \boldsymbol{\varepsilon}\to\boldsymbol{\varepsilon}}\,\text{VAR} \quad \{f:\boldsymbol{\varepsilon}\to\varepsilon;\ x:\beta\} \vdash x : \boldsymbol{\varepsilon}}{\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}\,\text{APP}}{\dfrac{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\,\text{FUN}}\,\text{FUN}}\,\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\boldsymbol{\{\zeta \equiv \varepsilon, \phi \equiv \varepsilon\}} \circ \{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Compose substitutions

$$
\dfrac{
  \dfrac{
    \dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\,\textbf{VAR}
    \qquad
    \dfrac{
      \dfrac{}{\{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash f : \varepsilon\to\varepsilon}\,\textbf{VAR}
      \qquad
      \{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash x : \varepsilon
    }{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}\,\textbf{APP}
  }{\{f:\delta\ ;\ x:\beta\} \vdash f\ (f\ x) : \varepsilon}\,\textbf{APP}
}{
  \dfrac{\{x:\beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\,\textbf{FUN}
}\,\textbf{FUN}
$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\zeta \equiv \varepsilon, \phi \equiv \varepsilon\} \circ \{\delta \equiv \phi \to \varepsilon\}$

Type Inference - Example

Compose substitutions

$$\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\,\text{VAR} \qquad \dfrac{\dfrac{}{\{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash f : \varepsilon\to\varepsilon}\,\text{VAR} \quad \{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash x : \varepsilon}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}\,\text{APP}}{\dfrac{\{f:\delta;\ x:\beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x:\beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\,\text{FUN}}\,\text{FUN}}\,\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\zeta \equiv \varepsilon,\ \phi \equiv \varepsilon,\ \delta \equiv \varepsilon \to \varepsilon\}$

Type Inference - Example

Recursively find substitution

$$\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash f : \varepsilon\to\varepsilon}\text{VAR} \quad \dfrac{???}{\{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash x : \varepsilon}}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}\text{APP}}{\dfrac{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma); \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\zeta \equiv \varepsilon, \phi \equiv \varepsilon, \delta \equiv \varepsilon \to \varepsilon\}$

Type Inference - Example

Variable rule

Decompose the expression into components

$$
\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\,\text{VAR} \quad \dfrac{\dfrac{}{\{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash f : \varepsilon\to\varepsilon}\,\text{VAR} \quad \dfrac{}{\{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash x : \varepsilon}\,\text{VAR}}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}\,\text{APP}}{\dfrac{\dfrac{\{f:\delta\ ;\ x:\beta\} \vdash f\ (f\ x) : \varepsilon}{\{x:\beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\,\text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\,\text{FUN}}\,\text{APP}
$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\zeta \equiv \varepsilon,\ \phi \equiv \varepsilon,\ \delta \equiv \varepsilon \to \varepsilon\}$

Type Inference - Example

Variable rule

Generate constraints
using the typing rule

$$\dfrac{\dfrac{}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon}\text{VAR} \qquad \dfrac{\dfrac{}{\{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash f : \varepsilon\to\varepsilon}\text{VAR} \quad \dfrac{}{\{f:\varepsilon\to\varepsilon;\ x:\boldsymbol{\beta}\} \vdash x : \boldsymbol{\varepsilon}}\text{VAR}}{\{f:\phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi}\text{APP}}{\dfrac{\{f:\delta\ ;\ x:\beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x:\beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon);\ \boldsymbol{\varepsilon \equiv \beta}$

Substituted: $\{\zeta \equiv \varepsilon,\ \phi \equiv \varepsilon,\ \delta \equiv \varepsilon \to \varepsilon\}$

Questions so far?

Example

Type Inference - Example

Variable rule

Generate constraints using the typing rule

$$\{f:\varepsilon\to\varepsilon;\ x:\beta\} \vdash f : \varepsilon\to\varepsilon \quad \text{VAR}$$

$$\{f:\varepsilon\to\varepsilon;\ x:\boldsymbol{\beta}\} \vdash x : \boldsymbol{\varepsilon} \quad \text{VAR}$$

$$\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f : \phi\to\varepsilon \quad \text{VAR}$$

$$\{f: \phi\to\varepsilon;\ x:\beta\} \vdash f\ x : \phi \quad \text{APP}$$

$$\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon \quad \text{APP}$$

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

$$\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha \quad \text{FUN}$$

Constraints: $\alpha \equiv (\beta \to \gamma)$; $\gamma \equiv (\delta \to \varepsilon)$; $\boldsymbol{\varepsilon \equiv \beta}$

Substituted: $\{\zeta \equiv \varepsilon, \phi \equiv \varepsilon, \delta \equiv \varepsilon \to \varepsilon\}$

Example

Type Inference - Example

Apply substitution
(since the RHS is done)

VAR — VAR

$\{f:\boldsymbol{\varepsilon \rightarrow \varepsilon};\ x:\beta\} \vdash f : \boldsymbol{\varepsilon \rightarrow \varepsilon}$ $\quad$ $\{f:\boldsymbol{\varepsilon \rightarrow \varepsilon};\ x:\beta\} \vdash x : \boldsymbol{\varepsilon}$

VAR — APP

$\{f: \phi \rightarrow \varepsilon;\ x:\beta\} \vdash f : \phi \rightarrow \varepsilon$ $\quad$ $\{f: \phi \rightarrow \varepsilon;\ x:\beta\} \vdash f\ x : \phi$

APP

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

FUN

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

FUN

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

Constraints: $\alpha \equiv (\beta \rightarrow \gamma);\ \gamma \equiv (\delta \rightarrow \varepsilon);\ \boldsymbol{\varepsilon \equiv \beta}$

Substituted: $\{\zeta \equiv \varepsilon,\ \phi \equiv \varepsilon,\ \delta \equiv \varepsilon \rightarrow \varepsilon\}$

Type Inference - Example

Apply substitution
(since the RHS is done)

VAR

$$\{f:\boldsymbol{\beta}\rightarrow\boldsymbol{\beta};\ x:\beta\} \vdash f : \boldsymbol{\beta}\rightarrow\boldsymbol{\beta}$$

VAR

$$\{f:\boldsymbol{\beta}\rightarrow\boldsymbol{\beta};\ x:\beta\} \vdash x : \boldsymbol{\beta}$$

VAR

$$\{f:\ \phi\rightarrow\varepsilon;\ x:\beta\} \vdash f : \phi\rightarrow\varepsilon$$

APP

$$\{f:\ \phi\rightarrow\varepsilon;\ x:\beta\} \vdash f\ x : \phi$$

APP

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

FUN

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

FUN

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

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

Substituted: $\boldsymbol{\{\varepsilon \equiv \beta\}} \circ \{\zeta \equiv \varepsilon,\ \phi \equiv \varepsilon,\ \delta \equiv \varepsilon \rightarrow \varepsilon\}$ Example

Type Inference - Example

Compose substitutions

$$\dfrac{\dfrac{}{\{f: \phi\to\varepsilon; x:\beta\} \vdash f : \phi\to\varepsilon}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\boldsymbol{\beta\to\beta}; x:\beta\} \vdash f : \boldsymbol{\beta\to\beta}}\text{VAR} \quad \dfrac{}{\{f:\boldsymbol{\beta\to\beta}; x:\beta\} \vdash x : \boldsymbol{\beta}}\text{VAR}}{\{f: \phi\to\varepsilon; x:\beta\} \vdash f\ x : \phi}\text{APP}}{\dfrac{\dfrac{\{f : \delta ; x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}\text{FUN}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma); \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\boldsymbol{\{\varepsilon \equiv \beta\} \circ \{\zeta \equiv \varepsilon, \phi \equiv \varepsilon, \delta \equiv \varepsilon \to \varepsilon\}}$

Type Inference - Example

Compose substitutions

VAR: $\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash f : \beta\rightarrow\beta$

VAR: $\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash x : \beta$

VAR: $\{f:\ \phi\rightarrow\varepsilon;\ x:\beta\} \vdash f : \phi\rightarrow\varepsilon$

APP: $\{f:\ \phi\rightarrow\varepsilon;\ x:\beta\} \vdash f\ x : \phi$

APP: $\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon$

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

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

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

Substituted: $\{\varepsilon \equiv \beta,\ \zeta \equiv \beta,\ \phi \equiv \beta,\ \delta \equiv \beta \rightarrow \beta\}$

Type Inference - Example

Done with that, so back down one more layer **apply substitution** there

$$\dfrac{\dfrac{}{\{f: \boldsymbol{\phi}\rightarrow\boldsymbol{\varepsilon};\ x:\beta\} \vdash f: \boldsymbol{\phi}\rightarrow\boldsymbol{\varepsilon}}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash f : \beta\rightarrow\beta}\text{VAR} \quad \dfrac{}{\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash x : \beta}\text{VAR}}{\{f: \boldsymbol{\phi}\rightarrow\boldsymbol{\varepsilon};\ x:\beta\} \vdash f\ x : \boldsymbol{\phi}}\text{APP}}{\dfrac{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

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

Substituted: $\{\boldsymbol{\varepsilon \equiv \beta},\ \zeta \equiv \beta,\ \boldsymbol{\phi \equiv \beta},\ \delta \equiv \beta \rightarrow \beta\}$ Example

Type Inference - Example

Done with that, so back down one more layer **apply substitution** there

$$\dfrac{\dfrac{}{\{f{:}\,\boldsymbol{\beta}\to\boldsymbol{\beta};\ x{:}\beta\} \vdash f{:}\ \boldsymbol{\beta}\to\boldsymbol{\beta}}\text{VAR} \quad \dfrac{\dfrac{}{\{f{:}\beta\to\beta;\ x{:}\beta\} \vdash f : \beta\to\beta}\text{VAR} \quad \dfrac{}{\{f{:}\beta\to\beta;\ x{:}\beta\} \vdash x : \beta}\text{VAR}}{\{f{:}\,\boldsymbol{\beta}\to\boldsymbol{\beta};\ x{:}\beta\} \vdash f\ x : \boldsymbol{\beta}}\text{APP}}{\dfrac{\{f : \delta\ ;\ x : \beta\} \vdash f\ (f\ x) : \varepsilon}{\dfrac{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \gamma}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\varepsilon \equiv \beta,\ \zeta \equiv \beta,\ \phi \equiv \beta,\ \delta \equiv \beta \to \beta\}$ Example

Type Inference - Example

Done with that, too, so back down one more layer **apply substitution** there

$$\dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash f:\beta\to\beta}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash f:\beta\to\beta}\text{VAR} \quad \dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash x:\beta}\text{VAR}}{\{f:\beta\to\beta;\ x:\beta\} \vdash f\ x:\beta}\text{APP}}{\dfrac{\{f:\boldsymbol{\delta};\ x:\beta\} \vdash f\ (f\ x):\boldsymbol{\varepsilon}}{\dfrac{\{x:\beta\} \vdash (\text{fun } f \to f\ (f\ x)):\gamma}{\{\ \} \vdash (\text{fun } x \to \text{fun } f \to f\ (f\ x)):\alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma);\ \gamma \equiv (\delta \to \varepsilon)$

Substituted: $\{\boldsymbol{\varepsilon \equiv \beta},\ \zeta \equiv \beta,\ \phi \equiv \beta,\ \boldsymbol{\delta \equiv \beta} \to \boldsymbol{\beta}\}$ Example

Type Inference - Example

Done with that, too, so back down one more layer **apply substitution** there

VAR VAR

$\{f:\beta\rightarrow\beta;\ x:\beta\}$ $\{f:\beta\rightarrow\beta;\ x:\beta\}$

$\vdash f : \beta\rightarrow\beta$ $\vdash x : \beta$

VAR APP

$\{f: \beta\rightarrow\beta;\ x:\beta\} \vdash f: \beta\rightarrow\beta$ $\{f: \beta\rightarrow\beta;\ x:\beta\} \vdash f\ x : \beta$

APP

$\{f : \boldsymbol{\beta}\rightarrow\boldsymbol{\beta}\ ;\ x : \beta\} \vdash f\ (f\ x) : \boldsymbol{\beta}$ FUN

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

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

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

Substituted: $\{\varepsilon \equiv \beta,\ \zeta \equiv \beta,\ \phi \equiv \beta,\ \delta \equiv \beta \rightarrow \beta\}$

Type Inference - Example

Done once more, so back down one more layer **apply substitution** there

VAR: $\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash f : \beta\rightarrow\beta$

VAR: $\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash x : \beta$

APP: $\{f: \beta\rightarrow\beta;\ x:\beta\} \vdash f\ x : \beta$

VAR: $\{f: \beta\rightarrow\beta;\ x:\beta\} \vdash f: \beta\rightarrow\beta$

APP: $\{f : \beta\rightarrow\beta\ ;\ x : \beta\} \vdash f\ (f\ x) : \beta$

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

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

Constraints: $\alpha \equiv (\beta \rightarrow \gamma)$; $\boldsymbol{\gamma \equiv (\delta \rightarrow \varepsilon)}$

Substituted: $\{\varepsilon \equiv \beta,\ \zeta \equiv \beta,\ \phi \equiv \beta,\ \delta \equiv \beta \rightarrow \beta\}$

Type Inference - Example

Done once more, so back down one more layer **apply substitution** there

VAR VAR

$\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash f : \beta\rightarrow\beta$ $\quad$ $\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash x : \beta$

VAR APP

$\{f: \beta\rightarrow\beta;\ x:\beta\} \vdash f: \beta\rightarrow\beta$ $\quad$ $\{f: \beta\rightarrow\beta;\ x:\beta\} \vdash f\ x : \beta$

APP

$\{f : \beta\rightarrow\beta\ ;\ x : \beta\} \vdash f\ (f\ x) : \beta$

FUN

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

FUN

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

Constraints: $\alpha \equiv (\beta \rightarrow \gamma)$; $\boldsymbol{\gamma \equiv (\delta \rightarrow \varepsilon)}$

Substituted: $\{\boldsymbol{\varepsilon \equiv \beta}, \zeta \equiv \beta, \phi \equiv \beta, \boldsymbol{\delta \equiv \beta \rightarrow \beta}\}$ Example

Type Inference - Example

Done once more, so back down one more layer **apply substitution** there

$$\dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\}\vdash f:\beta\to\beta}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\}\vdash f:\beta\to\beta}\text{VAR} \quad \dfrac{}{\{f:\beta\to\beta;\ x:\beta\}\vdash x:\beta}\text{VAR}}{\{f:\beta\to\beta;\ x:\beta\}\vdash f\ x:\beta}\text{APP}}{\dfrac{\{f:\beta\to\beta;\ x:\beta\}\vdash f\ (f\ x):\beta}{\dfrac{\{x:\beta\}\vdash (\text{fun } f \text{ -> } f\ (f\ x)):\boldsymbol{\gamma}}{\{\ \}\vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)):\alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma)$; $\boldsymbol{\gamma \equiv ((\beta \to \beta) \to \beta)}$

Substituted: $\{\varepsilon \equiv \beta, \zeta \equiv \beta, \phi \equiv \beta, \delta \equiv \beta \to \beta\}$

Type Inference - Example

Done once more, so back down one more layer **apply substitution** there

$$\dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash f:\beta\to\beta}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash f:\beta\to\beta}\text{VAR} \quad \dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash x:\beta}\text{VAR}}{\{f:\beta\to\beta;\ x:\beta\} \vdash f\ x:\beta}\text{APP}}{\dfrac{\{f:\beta\to\beta\ ;\ x:\beta\} \vdash f\ (f\ x):\beta}{\dfrac{\{x:\beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : \mathbf{((\beta \to \beta) \to \beta)}}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\alpha \equiv (\beta \to \gamma)$

Substituted: $\{\gamma \equiv ((\beta\to\beta)\to\beta),\ \varepsilon \equiv \beta,\ \ldots,\ \delta \equiv \beta\to\beta\}$ Example

Type Inference - Example

Back to the very start.

$$\dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash f:\beta\to\beta}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash f : \beta\to\beta}\text{VAR} \quad \dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash x : \beta}\text{VAR}}{\{f:\beta\to\beta;\ x:\beta\} \vdash f\ x : \beta}\text{APP}}{\dfrac{\{f:\beta\to\beta\ ;\ x:\beta\} \vdash f\ (f\ x) : \beta}{\dfrac{\{x:\beta\} \vdash (\text{fun } f \to f\ (f\ x)) : ((\beta\to\beta)\to\beta)}{\{\ \} \vdash (\text{fun } x \to \text{fun } f \to f\ (f\ x)) : \alpha}\text{FUN}}\text{FUN}}\text{APP}$$

Constraints: $\boldsymbol{\alpha \equiv (\beta \to \gamma)}$

Substituted: $\{\gamma \equiv ((\beta\to\beta)\to\beta),\ \varepsilon \equiv \beta,\ \ldots,\ \delta \equiv \beta\to\beta\}$ Example

Type Inference - Example

Back to the very start.

$$\dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash f:\beta\to\beta}\ \text{VAR} \qquad \dfrac{\dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash f:\beta\to\beta}\ \text{VAR} \quad \dfrac{}{\{f:\beta\to\beta;\ x:\beta\} \vdash x:\beta}\ \text{VAR}}{\{f:\beta\to\beta;\ x:\beta\} \vdash f\ x:\beta}\ \text{APP}}{\dfrac{\{f:\beta\to\beta\ ;\ x:\beta\} \vdash f\ (f\ x):\beta}{\dfrac{\{x:\beta\} \vdash (\text{fun } f \to f\ (f\ x)) : ((\beta\to\beta)\to\beta)}{\{\ \} \vdash (\text{fun } x \to \text{fun } f \to f\ (f\ x)) : \alpha}\ \text{FUN}}\ \text{FUN}}\ \text{APP}$$

Constraints: $\boldsymbol{\alpha \equiv (\beta \to ((\beta \to \beta) \to \beta))}$

Substituted: $\{\gamma \equiv ((\beta\to\beta)\to\beta),\ \varepsilon \equiv \beta,\ \ldots,\ \delta \equiv \beta\to\beta\}$ Example

Type Inference - Example

Now we substitute.

$$\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash f : \beta\rightarrow\beta \quad (\textbf{VAR})$$

$$\{f:\beta\rightarrow\beta;\ x:\beta\} \vdash x : \beta \quad (\textbf{VAR})$$

$$\{f: \beta\rightarrow\beta;\ x:\beta\} \vdash f: \beta\rightarrow\beta \quad (\textbf{VAR})$$

$$\{f: \beta\rightarrow\beta;\ x:\beta\} \vdash f\ x : \beta \quad (\textbf{APP})$$

$$\{f : \beta\rightarrow\beta\ ;\ x : \beta\} \vdash f\ (f\ x) : \beta \quad (\textbf{APP})$$

$$\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : ((\beta \rightarrow \beta) \rightarrow \beta) \quad (\textbf{FUN})$$

$$\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \boldsymbol{\alpha} \quad (\textbf{FUN})$$

Constraints: $\boldsymbol{\alpha \equiv (\beta \rightarrow ((\beta \rightarrow \beta) \rightarrow \beta))}$

Substituted: $\{\gamma \equiv ((\beta\rightarrow\beta)\rightarrow\beta),\ \varepsilon \equiv \beta,\ \ldots,\ \delta \equiv \beta\rightarrow\beta\}$

Type Inference - Example

Now we substitute.

$$\dfrac{\dfrac{}{\{f: \beta\to\beta; x:\beta\} \vdash f: \beta\to\beta}\text{VAR} \quad \dfrac{\dfrac{}{\{f:\beta\to\beta; x:\beta\} \vdash f : \beta\to\beta}\text{VAR} \quad \dfrac{}{\{f:\beta\to\beta; x:\beta\} \vdash x : \beta}\text{VAR}}{\{f: \beta\to\beta; x:\beta\} \vdash f\ x : \beta}\text{APP}}{\dfrac{\{f : \beta\to\beta ; x : \beta\} \vdash f\ (f\ x) : \beta}{\dfrac{\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : ((\beta \to \beta) \to \beta)}{\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : \boldsymbol{(\beta \to ((\beta \to \beta) \to \beta))}}\text{FUN}}\text{FUN}}\text{APP}$$

Substituted: $\{\alpha \equiv (\beta \to ((\beta \to \beta) \to \beta)), \gamma \equiv ((\beta \to \beta) \to \beta), \varepsilon \equiv \beta, \zeta \equiv \beta, \phi \equiv \beta, \delta \equiv \beta \to \beta\}$

Type Inference - Example

We are done.

VAR VAR

$\{f{:}\beta\rightarrow\beta;\ x{:}\beta\} \vdash f : \beta\rightarrow\beta$ $\quad$ $\{f{:}\beta\rightarrow\beta;\ x{:}\beta\} \vdash x : \beta$

VAR APP

$\{f{:}\ \beta\rightarrow\beta;\ x{:}\beta\} \vdash f{:}\ \beta\rightarrow\beta$ $\quad$ $\{f{:}\ \beta\rightarrow\beta;\ x{:}\beta\} \vdash f\ x : \beta$

APP

$\{f : \beta\rightarrow\beta\ ;\ x : \beta\} \vdash f\ (f\ x) : \beta$

FUN

$\{x : \beta\} \vdash (\text{fun } f \text{ -> } f\ (f\ x)) : ((\beta \rightarrow \beta) \rightarrow \beta)$

FUN

$\{\ \} \vdash (\text{fun } x \text{ -> fun } f \text{ -> } f\ (f\ x)) : (\beta \rightarrow ((\beta \rightarrow \beta) \rightarrow \beta))$

Substituted: $\{\alpha \equiv (\beta \rightarrow ((\beta \rightarrow \beta) \rightarrow \beta)),\ \gamma \equiv ((\beta \rightarrow \beta) \rightarrow \beta),$
$\varepsilon \equiv \beta,\ \zeta \equiv \beta,\ \phi \equiv \beta,\ \delta \equiv \beta \rightarrow \beta\}$

Example

Questions so far?

Type Inference Algorithm

Type Inference Algorithm

- Let **infer (Γ, e, T) = σ**
  - **Γ** is a **typing environment** (giving polymorphic types to expression variables)
  - **e** is an **expression**
  - **T** is a **type** (with type variables)
  - **σ** is a **substitution** of types for type variables
- Idea: **σ** is a **substitution** that **solves constraints** on type variables **necessary** for **Γ ⊢ e : T**
- Should have σ(Γ) ⊢ e : σ(T) valid

Type Inference Algorithm

- Let **infer (Γ, e, T) = σ**
  - **Γ** is a **typing environment** (giving polymorphic types to expression variables)
  - **e** is an **expression**
  - **T** is a **type** (with type variables)
  - **σ** is a **substitution** of types for type variables
- Idea: **σ** is a **substitution** that **solves constraints** on type variables **necessary** for **Γ ⊢ e : T**
- Should have σ(Γ) ⊢ e : σ(T) valid

Type Inference Algorithm

- Let **infer (Γ, e, T) = σ**
  - **Γ** is a **typing environment** (giving polymorphic types to expression variables)
  - **e** is an **expression**
  - **T** is a **type** (with type variables)
  - **σ** is a **substitution** of types for type variables
- Idea: **σ** is a **substitution** that **solves constraints** on type variables **necessary** for **Γ ⊢ e : T**
- Should have σ(Γ) ⊢ e : σ(T) valid

Pseudocaml

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  match e with
  | Var v ->
      (* replace quantified type vars by fresh ones *)
      Unify{T ≡ freshInstance(Γ(v))}
  | Const c ->
      (* where Γ ⊢ c : ϕ by the constant rules *)
      Unify{T ≡ freshInstance ϕ }
  | Fun (x, e) ->
      (* given α, β fresh *)
      let σ = infer (Γ, x : α) e β in
      Unify({σ(τ) ≡ σ(α → β)}) ∘ σ
```

Algorithm

Example of Var Rule

(* Type inference returns a substitutions σ *)
infer {x : ∀ ‘a.(‘a * ‘b) list, y : ‘b} x ((int * string) list) =>
Unify {((int * string) list) ≡ freshInstance(∀ ‘a.(‘a * ‘b) list)} =>
Unify {((int * string) list) ≡ (‘c * ‘b) list} =>
{‘c ≡ int, ‘b ≡ string} (* σ *)

(* We can substitute using σ to validate this *)
σ({x : ∀ ‘a.(‘a * ‘b) list, y : ‘b}) ⊢ x : σ((int * string) list)
{x : ∀ ‘a.(‘a * string) list, y : string} ⊢ x : (int * string) list
(* Note we have updated the environment *)

Algorithm

Example of Var Rule

(* Type inference returns a substitutions σ *)

infer {x : ∀ ‘a.(‘a * ‘b) list, y : ‘b} x ((int * string) list) =>

Unify {((int * string) list) ≡ freshInstance(∀ ‘a.(‘a * ‘b) list)} =>

Unify {((int * string) list) ≡ (‘c * ‘b) list} =>

{‘c ≡ int, ‘b ≡ string} (* σ *)

(* We can substitute using σ to validate this *)

σ({x : ∀ ‘a.(‘a * ‘b) list, y : ‘b}) ⊢ x : σ((int * string) list)

{x : ∀ ‘a.(‘a * string) list, y : string} ⊢ x : (int * string) list

(* Note we have updated the environment *)

Algorithm

Example of Var Rule

(* Type inference returns a substitutions σ *)
infer {x : ∀ ‘a.(‘a * ‘b) list, y : ‘b} x ((int * string) list) =>
Unify {((int * string) list) ≡ freshInstance(∀ ‘a.(‘a * ‘b) list)} =>
Unify {((int * string) list) ≡ (‘c * ‘b) list} =>
{‘c ≡ int, ‘b ≡ string} (* σ *)

(* We can substitute using σ to validate this *)
σ({x : ∀ ‘a.(‘a * ‘b) list, y : ‘b}) ⊢ x : σ((int * string) list)
{x : ∀ ‘a.(‘a * string) list, y : string} ⊢ x : (int * string) list
(* Note we have updated the environment *)

Algorithm

Example of Var Rule

(* Type inference returns a substitutions σ *)
infer {x : ∀ ‘a.(‘a * ‘b) list, y : ‘b} x ((int * string) list) =>
Unify {((int * string) list) ≡ freshInstance(∀ ‘a.(‘a * ‘b) list)} =>
Unify {((int * string) list) ≡ (‘c * ‘b) list} =>
{‘c ≡ int, ‘b ≡ string} (* σ *)

(* We can substitute using σ to validate this *)
σ({x : ∀ ‘a.(‘a * ‘b) list, y : ‘b}) ⊢ x : σ((int * string) list)
{x : ∀ ‘a.(‘a * string) list, y : string} ⊢ x : (int * string) list
(* Note we have updated the environment *)

Algorithm

Example of Var Rule

(* Type inference returns a substitutions σ *)
infer {x : ∀ ‘a.(‘a * ‘b) list, y : ‘b} x ((int * string) list) =>
Unify {((int * string) list) ≡ freshInstance(∀ ‘a.(‘a * ‘b) list)} =>
Unify {((int * string) list) ≡ (‘c * ‘b) list} =>
{‘c ≡ int, ‘b ≡ string} (* σ *)

(* We can substitute using σ to validate this *)
σ({x : ∀ ‘a.(‘a * ‘b) list, y : ‘b}) ⊢ x : σ((int * string) list)
{x : ∀ ‘a.(‘a * string) list, y : string} ⊢ x : (int * string) list
(* Note we have updated the environment *)

Example of Var Rule

(* Type inference returns a substitutions σ *)
infer {x : ∀ ‘a.(‘a * ‘b) list, y : ‘b} x ((int * string) list) =>
Unify {((int * string) list) ≡ freshInstance(∀ ‘a.(‘a * ‘b) list)} =>
Unify {((int * string) list) ≡ (‘c * ‘b) list} =>
{‘c ≡ int, ‘b ≡ string} (* σ *)

(* We can substitute using σ to validate this *)
σ({x : ∀ ‘a.(‘a * ‘b) list, y : ‘b}) ⊢ x : σ((int * string) list)
{x : ∀ ‘a.(‘a * string) list, y : string} ⊢ x : (int * string) list
(* Note we have updated the environment *)

Algorithm

Example of Var Rule

(* Type inference returns a substitutions σ *)
infer {x : ∀ 'a.('a * 'b) list, y : 'b} x ((int * string) list) =>
Unify {((int * string) list) ≡ freshInstance(∀ 'a.('a * 'b) list)} =>
Unify {((int * string) list) ≡ ('c * 'b) list} =>
{'c ≡ int, 'b ≡ string} (* σ *)

(* We can substitute using σ to validate this *)
σ({x : ∀ 'a.('a * 'b) list, y : 'b}) ⊢ x : σ((int * string) list)
{x : ∀ 'a.('a * string) list, y : string} ⊢ x : (int * string) list
(* Note we have updated the environment *)

Questions so far?

Pseudocaml

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  match e with
  | Var v ->
     (* replace quantified type vars by fresh ones *)
     Unify{T ≡ freshInstance(Γ(v))}
  | Const c ->
     (* where Γ ⊢ c : ϕ by the constant rules *)
     Unify{T ≡ freshInstance ϕ }
  | Fun (x, e) ->
     (* given α, β fresh *)
     let σ = infer (Γ, x : α) e β in
     Unify({σ(T) ≡ σ(α → β)}) ∘ σ
```

Algorithm

Pseudocaml

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  match e with
  | Var v ->
      (* replace quantified type vars by fresh ones *)
      Unify{T ≡ freshInstance(Γ(v))}
  | Const c ->
      (* where Γ ⊢ c : ϕ by the constant rules *)
      Unify{T ≡ freshInstance ϕ }
  | Fun (x, e) ->
      (* given α, β fresh *)
      let σ = infer (Γ, x : α) e β in
      Unify({σ(T) ≡ σ(α → β)}) ∘ σ
```

Algorithm

Pseudocaml

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  match e with
  | Var v ->
      (* replace quantified type vars by fresh ones *)
      Unify{T ≡ freshInstance(Γ(v))}
  | Const c ->
      (* where Γ ⊢ c : ϕ by the constant rules *)
      Unify{T ≡ freshInstance ϕ }
  | Fun (x, e) ->
      (* given α, β fresh *)
      let σ = infer (Γ, x : α) e β in
      Unify({σ(τ) ≡ σ(α → β)}) ∘ σ
```

Algorithm

Pseudocaml (continued)

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  …
  | App (e1, e2) ->
      (* Given α fresh *)
      let σ1 = infer Γ e1 (α → T) in
      let σ2 = infer (σ1(Γ)) e2 (σ1(α)) in
      σ2 ∘ σ1
  | If (e1, e2, e3) ->
      let σ1 = infer Γ e1 bool in
      let σ2 = infer (σ1(Γ)) e2 (σ1(T)) in
      let σ3 = infer ((σ2 ∘ σ1)(Γ)) e3 ((σ2 ∘ σ1)(T)) in
      σ3 ∘ σ2 ∘ σ1
```

Algorithm

Pseudocaml (continued)

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  …
  | App (e1, e2) ->
      (* Given α fresh *)
      let σ1 = infer Γ e1 (α → T) in
      let σ2 = infer (σ1(Γ)) e2 (σ1(α)) in
      σ2 ο σ1
  | If (e1, e2, e3) ->
      let σ1 = infer Γ e1 bool in
      let σ2 = infer (σ1(Γ)) e2 (σ1(T)) in
      let σ3 = infer ((σ2 ο σ1)(Γ)) e3 ((σ2 ο σ1)(T)) in
      σ3 ο σ2 ο σ1
```

Algorithm

Pseudocaml (continued)

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  …
  | Let (x, e1, e2) ->
      (* given α fresh *)
      let σ1 = infer Γ e1 α in
      let σ2 =
        infer((σ1(Γ)), x : GEN(σ1(Γ), σ1(α))) e2 (σ1(T))
      in σ2 o σ1
```

Pseudocaml (continued)

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  …
  | LetRec (x, e1, e2) ->
      (* given α fresh *)
      let σ1 = infer (Γ, x : α) e1 α in
      let σ2 =
        infer((σ1(Γ)), x : GEN(σ1(Γ), σ1(α))) e2 (σ1(T))
      in σ2 o σ1
```

Algorithm

Pseudocaml (continued)

let infer (Γ : env) (e: exp) (T : type) : **substitution** =
  … (* omitted *)

(* to infer a **type**, introduce type_of *)
let type_of Γ e =
  (* apply inferred substitution in fresh variable α *)
  let σ = infer Γ *e* α in σ(α)

Algorithm

Pseudocaml (continued)

```
let infer (Γ : env) (e: exp) (T : type) : substitution =
  … (* omitted *)

(* to infer a type, introduce type_of *)
let type_of Γ e =
  (* apply inferred substitution in fresh variable α *)
  let σ = infer Γ e α in σ(α)
```

Algorithm

Questions?

No Class Thursday due to Midterm!

Next Class

- **Midterm instead of class on Thursday**
- **Please sign up in CBTF** if you haven't yet
- **MP6 deadline extended**
- **EC2** after midterm
- All deadlines can be found on **course website**
- Use **office hours** and **class forums** for help