MTAT.05.105 Type Theory

Second-order λ-calculus

Second-order λ-calculus: types

- Syntax of types:

$$\begin{array}{rcll} \tau & ::= & \alpha & \text{type variable} \\ & | & \tau_1 \to \tau_2 & \text{function type} \\ & | & \forall \alpha . \tau & \text{polymorphic type} \end{array}$$

- Type constructor $\to$ is right associative.

$$\tau_1 \to \tau_2 \to \tau_3 \quad \equiv \quad \tau_1 \to (\tau_2 \to \tau_3)$$

- Universal quantification $\forall$ binds weakest.

$$\forall \alpha . \alpha \to \forall \beta . \beta \quad \equiv \quad \forall \alpha . (\alpha \to \forall \beta . \beta)$$

Second-order λ-calculus: terms

- Syntax of terms:

$$
\begin{array}{rcll}
e & ::= & x & \text{variable} \\
& | & e_1\, e_2 & \text{application} \\
& | & \lambda x : \tau .\, e & \text{abstraction} \\
& | & e\, \tau & \text{type application} \\
& | & \Lambda \alpha . e & \text{type abstraction}
\end{array}
$$

- For readability, we often write type annotations as superscripts.

Second-order λ-calculus: terms

- Syntax of terms:

$$\begin{array}{rcll} e & ::= & x & \text{variable} \\ & \mid & e_1\, e_2 & \text{application} \\ & \mid & \lambda x : \tau .\, e & \text{abstraction} \\ & \mid & e\, \tau & \text{type application} \\ & \mid & \Lambda \alpha . e & \text{type abstraction} \end{array}$$

- For readability, we often write type annotations as superscripts.
- Example:

$$dbl \equiv \Lambda \alpha . \lambda f^{\alpha \to \alpha} . \lambda x^{\alpha} . f(fx)$$

Second-order λ-calculus

- β-reduction rules:

$$\begin{aligned} (\lambda x{:}\tau.e_1)\, e_2 \quad &\rightarrow_\beta \quad e_1[x \mapsto e_2] \\ (\Lambda\alpha.e)\, \tau \quad &\rightarrow_\beta \quad e[\alpha \mapsto \tau] \end{aligned}$$

- Note that in $e[\alpha \mapsto \tau]$, free occurences of α are substituted both in type applications and type annotations.

Second-order λ-calculus

- β-reduction rules:

$$(\lambda x{:}\tau.e_1)\, e_2 \quad \to_\beta \quad e_1[x \mapsto e_2]$$

$$(\Lambda\alpha.e)\,\tau \quad \to_\beta \quad e[\alpha \mapsto \tau]$$

- Note that in $e[\alpha \mapsto \tau]$, free occurences of α are substituted both in type applications and type annotations.

- Example: Let $3 : \mathsf{Nat}$ and $\mathit{inc} : \mathsf{Nat} \to \mathsf{Nat}$

$$\mathit{dbl}\ \mathsf{Nat}\ \mathit{inc}\ 3$$

Second-order λ-calculus

- β-reduction rules:

$$(\lambda x{:}\tau.e_1)\,e_2 \quad \to_\beta \quad e_1[x \mapsto e_2]$$
$$(\Lambda\alpha.e)\,\tau \quad \to_\beta \quad e[\alpha \mapsto \tau]$$

- Note that in $e[\alpha \mapsto \tau]$, free occurences of α are substituted both in type applications and type annotations.

- Example: Let $3 : \mathsf{Nat}$ and $inc : \mathsf{Nat} \to \mathsf{Nat}$

$$dbl\ \mathsf{Nat}\ inc\ 3 \quad \equiv \quad (\Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx))\ \mathsf{Nat}\ inc\ 3$$

Second-order λ-calculus

- β-reduction rules:

$$(\lambda x{:}\tau.e_1)\, e_2 \quad \rightarrow_\beta \quad e_1[x \mapsto e_2]$$
$$(\Lambda\alpha.e)\, \tau \quad \rightarrow_\beta \quad e[\alpha \mapsto \tau]$$

- Note that in $e[\alpha \mapsto \tau]$, free occurences of α are substituted both in type applications and type annotations.

- Example: Let $3 : \text{Nat}$ and $inc : \text{Nat} \rightarrow \text{Nat}$

$$\begin{aligned} dbl\ \text{Nat}\ inc\ 3 \quad &\equiv \quad (\Lambda\alpha.\lambda f^{\alpha\rightarrow\alpha}.\lambda x^{\alpha}.f(fx))\ \text{Nat}\ inc\ 3 \\ &\rightarrow_\beta \quad (\lambda f^{\text{Nat}\rightarrow\text{Nat}}.\lambda x^{\text{Nat}}.f(fx))\ inc\ 3 \end{aligned}$$

Second-order λ-calculus

- β-reduction rules:

$$
\begin{array}{rcl}
(\lambda x{:}\tau.e_1)\,e_2 & \to_\beta & e_1[x \mapsto e_2] \\
(\Lambda\alpha.e)\,\tau & \to_\beta & e[\alpha \mapsto \tau]
\end{array}
$$

- Note that in $e[\alpha \mapsto \tau]$, free occurences of α are substituted both in type applications and type annotations.

- Example: Let $3 : \mathrm{Nat}$ and $inc : \mathrm{Nat} \to \mathrm{Nat}$

$$
\begin{array}{rcl}
dbl\ \mathrm{Nat}\ inc\ 3 & \equiv & (\Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx))\ \mathrm{Nat}\ inc\ 3 \\
 & \to_\beta & (\lambda f^{\mathrm{Nat}\to\mathrm{Nat}}.\lambda x^{\mathrm{Nat}}.f(fx))\ inc\ 3 \\
 & \to_\beta & (\lambda x^{\mathrm{Nat}}.inc(inc\, x))\, 3
\end{array}
$$

Second-order λ-calculus

- β-reduction rules:

$$\begin{aligned}(\lambda x{:}\tau.e_1)\,e_2 &\to_\beta e_1[x \mapsto e_2]\\ (\Lambda\alpha.e)\,\tau &\to_\beta e[\alpha \mapsto \tau]\end{aligned}$$

- Note that in $e[\alpha \mapsto \tau]$, free occurences of α are substituted both in type applications and type annotations.
- Example: Let $3 : \mathrm{Nat}$ and $inc : \mathrm{Nat} \to \mathrm{Nat}$

$$\begin{aligned}dbl\ \mathrm{Nat}\ inc\ 3 &\equiv (\Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx))\ \mathrm{Nat}\ inc\ 3\\ &\to_\beta (\lambda f^{\mathrm{Nat}\to\mathrm{Nat}}.\lambda x^{\mathrm{Nat}}.f(fx))\ inc\ 3\\ &\to_\beta (\lambda x^{\mathrm{Nat}}.inc(inc\ x))\,3\\ &\to_\beta inc(inc\ 3))\end{aligned}$$

Second-order λ-calculus: typing rules

- Typing rules for second-order λ-calculus:

$$\frac{}{\Gamma, x:\tau \vdash x:\tau}$$

$$\frac{\Gamma, x:\sigma \vdash e:\tau}{\Gamma \vdash \lambda x:\sigma.\, e:\sigma \to \tau}\ (x \notin \mathrm{dom}(\Gamma)) \qquad \frac{\Gamma \vdash e_1 : \sigma \to \tau \quad \Gamma \vdash e_2 : \sigma}{\Gamma \vdash e_1\, e_2 : \tau}$$

$$\frac{\Gamma \vdash e:\tau}{\Gamma \vdash (\Lambda\alpha.e) : \forall\alpha.\tau}\ (\alpha \notin \mathrm{FV}(\Gamma)) \qquad \frac{\Gamma \vdash e : \forall\alpha.\tau}{\Gamma \vdash e\,\sigma : \tau[\alpha \mapsto \sigma]}$$

- Further on we refer to the second-order λ-calculus as $\lambda 2$.

Second-order λ-calculus: examples

- The polymorhic identity:

$$id \quad \equiv \quad \Lambda\alpha.\lambda x^{\alpha}.x \qquad : \quad \forall\alpha.\alpha \to \alpha$$

Second-order λ-calculus: examples

- The polymorhic identity:

$$id \quad \equiv \quad \Lambda\alpha.\lambda x^{\alpha}.x \quad : \quad \forall\alpha.\alpha \to \alpha$$

- Typing derivation:

$$\cfrac{\cfrac{\overline{\{x{:}\alpha\} \vdash x : \alpha}}{\vdash \lambda x^{\alpha}.x : \alpha \to \alpha}}{\vdash \Lambda\alpha.\lambda x^{\alpha}.x : \forall\alpha.\alpha \to \alpha}$$

Second-order λ-calculus: examples

- The polymorhic identity:

$$id \quad \equiv \quad \Lambda\alpha.\lambda x^{\alpha}.x \quad : \quad \forall\alpha.\alpha \to \alpha$$

- Typing derivation:

$$\dfrac{\dfrac{\dfrac{}{\{x{:}\alpha\} \vdash x : \alpha}}{\vdash \lambda x^{\alpha}.x : \alpha \to \alpha}}{\vdash \Lambda\alpha.\lambda x^{\alpha}.x : \forall\alpha.\alpha \to \alpha}$$

- For any type τ and term $e : \tau$, we have:

$$\begin{array}{lcl} id\,\tau & : & \tau \to \tau \\ id\,\tau\,e & : & \tau \end{array}$$

Second-order λ-calculus: examples

- The polymorhic identity:

$$id \quad \equiv \quad \Lambda\alpha.\lambda x^{\alpha}.x \quad : \quad \forall\alpha.\alpha \rightarrow \alpha$$

- Typing derivation:

$$\cfrac{\cfrac{\overline{\{x{:}\alpha\} \vdash x : \alpha}}{\vdash \lambda x^{\alpha}.x : \alpha \rightarrow \alpha}}{\vdash \Lambda\alpha.\lambda x^{\alpha}.x : \forall\alpha.\alpha \rightarrow \alpha}$$

- For any type τ and term $e : \tau$, we have:

$$\begin{array}{lcl} id\,\tau & : & \tau \rightarrow \tau \\ id\,\tau\,e & : & \tau \end{array}$$

- In particular, if $\tau = \forall\alpha.\alpha \rightarrow \alpha$ and $e = id$, we have:

$$\begin{array}{lcl} id\,(\forall\alpha.\alpha \rightarrow \alpha) & : & (\forall\alpha.\alpha \rightarrow \alpha) \rightarrow (\forall\alpha.\alpha \rightarrow \alpha) \\ id\,(\forall\alpha.\alpha \rightarrow \alpha)\,id & : & \forall\alpha.\alpha \rightarrow \alpha \end{array}$$

Second-order λ-calculus: examples

- Doubling combinator:

$$dbl \quad \equiv \quad \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) \quad : \quad \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)$$

Second-order λ-calculus: examples

- Doubling combinator:

$$dbl \quad \equiv \quad \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) \quad : \quad \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\frac{}{\vdash \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) : \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Doubling combinator:

$$dbl \equiv \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) \quad : \quad \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\cfrac{\overline{\vdash \lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) : (\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) : \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Doubling combinator:

$$dbl \quad \equiv \quad \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) \quad : \quad \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\overline{\{f{:}\alpha\to\alpha\}\vdash\lambda x^{\alpha}.f(fx):\alpha\to\alpha}}{\vdash\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx):(\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash\Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx):\forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Doubling combinator:

$$dbl \quad \equiv \quad \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) \quad : \quad \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{f{:}\alpha\to\alpha,\, x{:}\alpha\}\vdash f(fx):\alpha}}{\{f{:}\alpha\to\alpha\}\vdash \lambda x^{\alpha}.f(fx):\alpha\to\alpha}}{\vdash \lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx):(\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx):\forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Doubling combinator:

$$dbl \quad \equiv \quad \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) \quad : \quad \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma\vdash f:\alpha\to\alpha}\qquad\dfrac{}{\Gamma\vdash fx:\alpha}}{\{f{:}\alpha\to\alpha,\ x{:}\alpha\}\vdash f(fx):\alpha}}{\{f{:}\alpha\to\alpha\}\vdash\lambda x^{\alpha}.f(fx):\alpha\to\alpha}}{\vdash\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx):(\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash\Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx):\forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Doubling combinator:

$$dbl \quad \equiv \quad \Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx) \quad : \quad \forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma\vdash f:\alpha\to\alpha} \qquad \dfrac{\dfrac{}{\Gamma\vdash f:\alpha\to\alpha} \qquad \dfrac{}{\Gamma\vdash x:\alpha}}{\Gamma\vdash fx:\alpha}}{\{f{:}\alpha\to\alpha,\ x{:}\alpha\}\vdash f(fx):\alpha}}{\{f{:}\alpha\to\alpha\}\vdash\lambda x^{\alpha}.f(fx):\alpha\to\alpha}}{\vdash\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx):(\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash\Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f(fx):\forall\alpha.(\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application:

$$\omega \equiv \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x$$
$$: (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$$

Second-order λ-calculus: examples

- Self-application:

$$\omega \equiv \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x$$
$$: (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{}{\vdash \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application:

$$\omega \equiv \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x \\ : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\overline{\{x{:}\forall\alpha.\alpha\to\alpha\}\vdash x\,(\forall\alpha.\alpha\to\alpha)\,x : (\forall\alpha.\alpha\to\alpha)}}{\vdash \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application:

$$\omega \quad \equiv \quad \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x$$
$$:(\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\overline{\Gamma\vdash x\,(\forall\alpha.\alpha\to\alpha):(\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)} \qquad \overline{\Gamma\vdash x:\forall\alpha.\alpha\to\alpha}}{\{x{:}\forall\alpha.\alpha\to\alpha\}\vdash x\,(\forall\alpha.\alpha\to\alpha)\,x:(\forall\alpha.\alpha\to\alpha)}}{\vdash\lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x:(\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application:

$$\omega \quad \equiv \quad \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x$$
$$: (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\overline{\Gamma\vdash x\,(\forall\alpha.\alpha\to\alpha) : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)} \qquad \overline{\Gamma\vdash x : \forall\alpha.\alpha\to\alpha}}{\{x{:}\forall\alpha.\alpha\to\alpha\}\vdash x\,(\forall\alpha.\alpha\to\alpha)\,x : (\forall\alpha.\alpha\to\alpha)}}{\vdash \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application:

$$\omega \equiv \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\dfrac{}{\Gamma\vdash x:\forall\alpha.\alpha\to\alpha}}{\Gamma\vdash x\,(\forall\alpha.\alpha\to\alpha):(\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)}\qquad\dfrac{}{\Gamma\vdash x:\forall\alpha.\alpha\to\alpha}}{\dfrac{\{x{:}\forall\alpha.\alpha\to\alpha\}\vdash x\,(\forall\alpha.\alpha\to\alpha)\,x:(\forall\alpha.\alpha\to\alpha)}{\vdash\lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x:(\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)}}$$

Second-order λ-calculus: examples

- Self-application:

$$\begin{aligned}\omega \quad \equiv \quad & \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x \\ & : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)\end{aligned}$$

- What about double self-application $\omega\,\omega$?

Second-order λ-calculus: examples

- Self-application:

$$\omega \equiv \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x$$
$$: (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$$

- What about double self-application $\omega\,\omega$?
  - Let $\Gamma = \{\omega : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)\}$

$$\frac{}{\Gamma\vdash\omega\,\omega:\tau}$$

Second-order λ-calculus: examples

- Self-application:

$$\omega \quad \equiv \quad \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha \to \alpha)\,x$$
$$: (\forall\alpha.\alpha \to \alpha) \to (\forall\alpha.\alpha \to \alpha)$$

- What about double self-application $\omega\,\omega$?
  - Let $\Gamma = \{\omega : (\forall\alpha.\alpha \to \alpha) \to (\forall\alpha.\alpha \to \alpha)\}$

$$\dfrac{\dfrac{}{\Gamma \vdash \omega : \sigma \to \tau} \qquad \dfrac{}{\Gamma \vdash \omega : \sigma}}{\Gamma \vdash \omega\,\omega : \tau}$$

Second-order λ-calculus: examples

- Self-application:

$$\omega \equiv \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha \to \alpha)\,x$$
$$: (\forall\alpha.\alpha \to \alpha) \to (\forall\alpha.\alpha \to \alpha)$$

- What about double self-application $\omega\,\omega$?
  - Let $\Gamma = \{\omega : (\forall\alpha.\alpha \to \alpha) \to (\forall\alpha.\alpha \to \alpha)\}$

$$\frac{\overline{\Gamma \vdash \omega : \sigma \to \tau} \qquad \overline{\Gamma \vdash \omega : \sigma}}{\Gamma \vdash \omega\,\omega : \tau}$$

  - Hence $\sigma = \forall\alpha.\alpha \to \alpha$ and $\tau = \forall\alpha.\alpha \to \alpha$.

Second-order λ-calculus: examples

- Self-application:

$$\omega \;\equiv\; \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha\to\alpha)\,x$$
$$: (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$$

- What about double self-application $\omega\,\omega$?
  - Let $\Gamma = \{\omega : (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)\}$

$$\dfrac{\dfrac{}{\Gamma\vdash\omega:\sigma\to\tau} \qquad \dfrac{}{\Gamma\vdash\omega:\sigma}}{\Gamma\vdash\omega\,\omega:\tau}$$

  - Hence $\sigma = \forall\alpha.\alpha\to\alpha$ and $\tau = \forall\alpha.\alpha\to\alpha$.
  - Hence $\sigma = (\forall\alpha.\alpha\to\alpha)\to(\forall\alpha.\alpha\to\alpha)$.

Second-order λ-calculus: examples

- Self-application:

$$\omega \equiv \lambda x^{\forall\alpha.\alpha\to\alpha}.x\,(\forall\alpha.\alpha \to \alpha)\,x$$
$$: (\forall\alpha.\alpha \to \alpha) \to (\forall\alpha.\alpha \to \alpha)$$

- What about double self-application $\omega\,\omega$?
  - Let $\Gamma = \{\omega : (\forall\alpha.\alpha \to \alpha) \to (\forall\alpha.\alpha \to \alpha)\}$

$$\frac{\overline{\Gamma \vdash \omega : \sigma \to \tau} \qquad \overline{\Gamma \vdash \omega : \sigma}}{\Gamma \vdash \omega\,\omega : \tau}$$

  - Hence $\sigma = \forall\alpha.\alpha \to \alpha$ and $\tau = \forall\alpha.\alpha \to \alpha$.
  - Hence $\sigma = (\forall\alpha.\alpha \to \alpha) \to (\forall\alpha.\alpha \to \alpha)$.
  - But these types are incompatible!

Second-order λ-calculus: examples

- Self-application (2):

$$\begin{aligned}\omega' \;\equiv\; & \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha) \\ & : \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)\end{aligned}$$

Second-order λ-calculus: examples

- Self-application (2):

$$\begin{aligned}\omega' \quad &\equiv \quad \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha) \\ &\qquad\qquad : \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)\end{aligned}$$

- Typing derivation:

$$\frac{}{\vdash \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha) : \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application (2):

$$\omega' \equiv \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha)$$
$$: \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{}{\vdash \lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha) : (\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha) : \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application (2):

$$\omega' \equiv \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha)$$
$$: \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\overline{\{x{:}\forall\alpha.\alpha\to\alpha\}\vdash x(\alpha\to\alpha)(x\alpha):\alpha\to\alpha}}{\vdash \lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha):(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha):\forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application (2):

$$\omega' \equiv \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha)$$
$$: \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\dfrac{\overline{\Gamma\vdash x(\alpha\to\alpha) : (\alpha\to\alpha)\to(\alpha\to\alpha)} \qquad \overline{\Gamma\vdash x\alpha : \alpha\to\alpha}}{\{x{:}\forall\alpha.\alpha\to\alpha\}\vdash x(\alpha\to\alpha)(x\alpha) : \alpha\to\alpha}}{\vdash \lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha) : (\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha) : \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application (2):

$$\begin{aligned}\omega' \quad &\equiv \quad \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha)\\ &\qquad\qquad : \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)\end{aligned}$$

- Typing derivation:

$$\dfrac{\dfrac{\dfrac{}{\Gamma\vdash x(\alpha\to\alpha):(\alpha\to\alpha)\to(\alpha\to\alpha)} \qquad \dfrac{\dfrac{}{\Gamma\vdash x:\forall\alpha.\alpha\to\alpha}}{\Gamma\vdash x\alpha:\alpha\to\alpha}}{\dfrac{\{x{:}\forall\alpha.\alpha\to\alpha\}\vdash x(\alpha\to\alpha)(x\alpha):\alpha\to\alpha}{\vdash\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha):(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}}}{\vdash\Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha):\forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Second-order λ-calculus: examples

- Self-application (2):

$$\omega' \equiv \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha) : \forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)$$

- Typing derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\Gamma\vdash x:\forall\alpha.\alpha\to\alpha}}{\Gamma\vdash x(\alpha\to\alpha):(\alpha\to\alpha)\to(\alpha\to\alpha)} \quad \dfrac{\overline{\Gamma\vdash x:\forall\alpha.\alpha\to\alpha}}{\Gamma\vdash x\alpha:\alpha\to\alpha}}{\{x{:}\forall\alpha.\alpha\to\alpha\}\vdash x(\alpha\to\alpha)(x\alpha):\alpha\to\alpha}}{\vdash \lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha):(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}}{\vdash \Lambda\alpha.\lambda x^{\forall\alpha.\alpha\to\alpha}.x(\alpha\to\alpha)(x\alpha):\forall\alpha.(\forall\alpha.\alpha\to\alpha)\to(\alpha\to\alpha)}$$

Properties of $\lambda 2$

- There is a Curry-Howard correspondence between $\lambda 2$ and $ND(\supset, \forall^2)$.
- Corollary (Type Inhabitation): Given a type τ, the problem of finding e such that $\vdash e : \tau$ is undecidable.
- Uniqueness of types:

  If $\Gamma \vdash e : \tau_1$ and $\Gamma \vdash e : \tau_2$, then $\tau_1 = \tau_2$.
- Decidability of typing: In $\lambda 2$, type checking and type inference are equivalent and decidable problems.
  - Type checking: Given Γ, e and τ, check whether the judgement $\Gamma \vdash e : \tau$ is derivable.
  - Type inference: Given Γ and e, find a type τ such that $\Gamma \vdash e : \tau$ is derivable or tell *"No"* if there is no such type.

Properties of $\lambda 2$

- Church-Rosser:

  If $e_0 \twoheadrightarrow_\beta e_1$ and $e_0 \twoheadrightarrow_\beta e_2$, then there is a term e_3 such that $e_1 \twoheadrightarrow_\beta e_3$ and $e_2 \twoheadrightarrow_\beta e_3$.

- Subject Reduction:

  If $\Gamma \vdash e_1 : \tau$ and $e_1 \twoheadrightarrow_\beta e_2$, then $\Gamma \vdash e_2 : \tau$.

- Strong Normalization:

  In $\lambda 2$, every β-reduction sequence is finite.

- Corollary: Fixpoint operators are not definable in $\lambda 2$.

Programming in $\lambda 2$: booleans

- Definition:

$$
\begin{array}{lcl}
\mathsf{Bool} & \equiv & \forall\alpha.\alpha \to \alpha \to \alpha \\
\mathtt{true} & \equiv & \Lambda\alpha.\lambda x^{\alpha}, y^{\alpha}.x : \mathsf{Bool} \\
\mathtt{false} & \equiv & \Lambda\alpha.\lambda x^{\alpha}, y^{\alpha}.y : \mathsf{Bool} \\
\mathbf{if}_{\tau}\, e_0\, \mathbf{then}\, e_1\, \mathbf{else}\, e_2 & \equiv & e_0\, \tau\, e_1\, e_2
\end{array}
$$

- Typing:

$$
\frac{\Gamma \vdash e_0 : \mathsf{Bool} \quad \Gamma \vdash e_1 : \tau \quad \Gamma \vdash e_2 : \tau}{\Gamma \vdash \mathbf{if}_{\tau}\, e_0\, \mathbf{then}\, e_1\, \mathbf{else}\, e_2 : \tau}
$$

- Reduction:

$$
\frac{e_0 \twoheadrightarrow \mathtt{true}}{\mathbf{if}_{\tau}\, e_0\, \mathbf{then}\, e_1\, \mathbf{else}\, e_2 \twoheadrightarrow e_1}
\qquad
\frac{e_0 \twoheadrightarrow \mathtt{false}}{\mathbf{if}_{\tau}\, e_0\, \mathbf{then}\, e_1\, \mathbf{else}\, e_2 \twoheadrightarrow e_2}
$$

Programming in $\lambda 2$: products

- Definition:

$$\begin{array}{lcl}
\tau_1 \times \tau_2 & \equiv & \forall \alpha.(\tau_1 \to \tau_2 \to \alpha) \to \alpha \\
(e_1, e_2) & \equiv & \Lambda \alpha. \lambda f^{\tau_1 \to \tau_2 \to \alpha}. f\, e_1\, e_2 \\
\mathtt{fst} & \equiv & \lambda p^{\tau_1 \times \tau_2}. p\, \tau_1\, (\lambda x^{\tau_1}, y^{\tau_2}. x) : \tau_1 \times \tau_2 \to \tau_1 \\
\mathtt{snd} & \equiv & \lambda p^{\tau_1 \times \tau_2}. p\, \tau_2\, (\lambda x^{\tau_1}, y^{\tau_2}. y) : \tau_1 \times \tau_2 \to \tau_2
\end{array}$$

- Typing:

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2}$$

- Reduction:

$$\begin{array}{lcl}
\mathtt{fst}\, (e_1, e_2) & \twoheadrightarrow & e_1 \\
\mathtt{snd}\, (e_1, e_2) & \twoheadrightarrow & e_2
\end{array}$$

Programming in $\lambda 2$: sums

- Definition:

$$
\begin{array}{lcl}
\tau_1 + \tau_2 & \equiv & \forall\alpha.(\tau_1 \to \alpha) \to (\tau_2 \to \alpha) \to \alpha \\
\mathtt{inl}\, e & \equiv & \Lambda\alpha.\lambda f^{\tau_1 \to \alpha}, g^{\tau_2 \to \alpha}.f\, e \\
\mathtt{inr}\, e & \equiv & \Lambda\alpha.\lambda f^{\tau_1 \to \alpha}, g^{\tau_2 \to \alpha}.g\, e \\
\mathbf{case}_\sigma\,(e_0;\, x_1^{\tau_1}.e_1;\, x_2^{\tau_2}.e_2) & \equiv & e_0\, \sigma\, (\lambda x_1^{\tau_1}.e_1)\, (\lambda x_2^{\tau_2}.e_2)
\end{array}
$$

- Typing:

$$
\frac{\Gamma \vdash e : \tau_1}{\Gamma \vdash \mathtt{inl}\, e : \tau_1 + \tau_2} \qquad \frac{\Gamma \vdash e : \tau_2}{\Gamma \vdash \mathtt{inr}\, e : \tau_1 + \tau_2}
$$

$$
\frac{\Gamma \vdash e_0 : \tau_1 + \tau_2 \quad \Gamma, x_1 : \tau_1 \vdash e_1 : \sigma \quad \Gamma, x_2 : \tau_2 \vdash e_2 : \sigma}{\Gamma \vdash \mathbf{case}_\sigma\,(e_0;\, x_1^{\tau_1}.e_1;\, x_2^{\tau_2}.e_2) : \sigma}
$$

- Reduction:

$$
\begin{array}{lcl}
\mathbf{case}_\tau\,(\mathtt{inl}\, e_0;\, x_1^{\tau_1}.e_1;\, x_2^{\tau_2}.e_2) & \twoheadrightarrow & e_1[x_1 \mapsto e_0] \\
\mathbf{case}_\tau\,(\mathtt{inr}\, e_0;\, x_1^{\tau_1}.e_1;\, x_2^{\tau_2}.e_2) & \twoheadrightarrow & e_2[x_2 \mapsto e_0]
\end{array}
$$

Programming in $\lambda 2$: finite types

- Definition:

$$\mathrm{Fin}_n \equiv \forall\alpha.\,\underbrace{\alpha \to \ldots \to \alpha}_{n \text{ times}} \to \alpha$$

$$\mathbf{e}_1 \equiv \Lambda\alpha.\lambda x_1^\alpha, \ldots, x_n^\alpha.x_1 : \mathrm{Fin}_n$$

$$\ldots$$

$$\mathbf{e}_n \equiv \Lambda\alpha.\lambda x_1^\alpha, \ldots, x_n^\alpha.x_n : \mathrm{Fin}_n$$

$$\mathbf{nCase}_\tau^n\,(e_0;\, e_1; \ldots;\, e_n) \equiv e_0\,\tau\, e_1 \ldots\, e_n$$

Programming in $\lambda 2$: finite types

- Definition:

$$\mathrm{Fin}_n \equiv \forall \alpha.\, \underbrace{\alpha \to \ldots \to \alpha}_{n \text{ times}} \to \alpha$$

$$\mathbf{e}_1 \equiv \Lambda \alpha . \lambda x_1^\alpha, \ldots, x_n^\alpha . x_1 : \mathrm{Fin}_n$$

$$\ldots$$

$$\mathbf{e}_n \equiv \Lambda \alpha . \lambda x_1^\alpha, \ldots, x_n^\alpha . x_n : \mathrm{Fin}_n$$

$$\mathbf{nCase}_\tau^n\,(e_0;\, e_1; \ldots;\, e_n) \equiv e_0\, \tau\, e_1 \ldots e_n$$

- In particular:

$$\begin{array}{lllll} \mathrm{Fin}_2 & \equiv & \forall \alpha . \alpha \to \alpha \to \alpha & \equiv & \mathrm{Bool} \\ \mathrm{Fin}_1 & \equiv & \forall \alpha . \alpha \to \alpha & \equiv & \mathrm{Unit} \\ \mathrm{Fin}_0 & \equiv & \forall \alpha . \alpha & \equiv & \mathrm{Void} \end{array}$$

Programming in $\lambda 2$: natural numbers

- Definition:

$$
\begin{array}{lcl}
\mathsf{Nat} & \equiv & \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha \\
\underline{0} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}.x : \mathsf{Nat} \\
\underline{1} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}.f\,x : \mathsf{Nat} \\
\underline{2} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}.f(f\,x) : \mathsf{Nat} \\
 & \dots & \\
\underline{n} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}.\underbrace{f(\dots(f}_{n\ \text{times}}\ x)\dots) : \mathsf{Nat} \\
\textbf{iter} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}, n^{\mathsf{Nat}}.n\,\alpha\,f\,x \\
 & & \quad : \forall\alpha.(\alpha \to \alpha) \to \alpha \to \mathsf{Nat} \to \alpha
\end{array}
$$

Programming in $\lambda 2$: natural numbers

- Definition:

$$
\begin{array}{lcl}
\mathsf{Nat} & \equiv & \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha \\
\underline{0} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}.x : \mathsf{Nat} \\
\underline{1} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}.f\ x : \mathsf{Nat} \\
\underline{2} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}.f(f\ x) : \mathsf{Nat} \\
 & \ldots & \\
\underline{n} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}.\underbrace{f(\ldots(f}_{n\ \text{times}}\ x)\ldots) : \mathsf{Nat} \\
\mathbf{iter} & \equiv & \Lambda\alpha.\lambda f^{\alpha\to\alpha}, x^{\alpha}, n^{\mathsf{Nat}}.n\ \alpha\ f\ x \\
 & & \quad : \forall\alpha.(\alpha \to \alpha) \to \alpha \to \mathsf{Nat} \to \alpha
\end{array}
$$

- Combinator **iter** is an iterator (or recursor) for naturals.

$$
\begin{array}{lcl}
\mathbf{iter}\,\tau\ s\ z\ \underline{0} & = & z \\
\mathbf{iter}\,\tau\ s\ z\ \underline{n+1} & = & s\,(\mathbf{iter}\,\tau\ s\ z\ \underline{n})
\end{array}
$$