

Types

CST Part II Paper 8 & 9

Victor Zhao
xz398@cam.ac.uk

1 Simply-Typed λ -Calculus

Syntax

Types	$T ::= 1 \mid 0 \mid T_1 \times T_2 \mid T_1 + T_2 \mid T_1 \rightarrow T_2$
Terms	$e ::= x \mid \langle \rangle \mid \langle e_1, e_2 \rangle \mid \text{fst } e \mid \text{snd } e \mid \text{L } e \mid \text{R } e \mid \text{case}(e, \text{L } x \rightarrow e_1, \text{R } y \rightarrow e_2)$ $\mid \lambda x : T. e \mid e_1 e_2 \mid \text{abort}$
Values	$v ::= \langle \rangle \mid \langle v_1, v_2 \rangle \mid \lambda x : T. e \mid \text{L } v \mid \text{R } v$
Contexts	$\Gamma ::= \cdot \mid \Gamma, x : T$

Typing rules

(I: introduction rule, E: elimination rule, HYP: hypothesis)

$$\begin{array}{c}
\frac{}{\Gamma \vdash \langle \rangle : 1} \text{1I} \quad \frac{\Gamma \vdash e_1 : T_1 \quad \Gamma \vdash e_2 : T_2}{\Gamma \vdash \langle e_1, e_2 \rangle : 1} \times\text{I} \quad \frac{\Gamma \vdash e : T_1 \times T_2}{\Gamma \vdash \text{fst } e : T_1} \times\text{E}_1 \quad \frac{\Gamma \vdash e : T_1 \times T_2}{\Gamma \vdash \text{snd } e : T_2} \times\text{E}_2 \\
\\
\frac{x : T \in \Gamma}{\Gamma \vdash x : T} \text{HYP} \quad \frac{\Gamma, x : T \vdash e : T'}{\Gamma \vdash \lambda x : T. e : T \rightarrow T'} \rightarrow\text{I} \quad \frac{\Gamma \vdash e_1 : T \rightarrow T' \quad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1 e_2 : T'} \rightarrow\text{E} \\
\\
\frac{\Gamma \vdash e : T_1}{\Gamma \vdash \text{L } e : T_1 + T_2} +\text{I}_1 \quad \frac{\Gamma \vdash e : T_2}{\Gamma \vdash \text{R } e : T_1 + T_2} +\text{I}_2 \\
\\
\frac{\Gamma \vdash e : T_1 + T_2 \quad \Gamma, x : X \vdash e_1 : T \quad \Gamma, x : X \vdash e_2 : T}{\Gamma \vdash \text{case}(e, \text{L } x \rightarrow e_1, \text{R } y \rightarrow e_2) : T} +\text{E} \\
\\
\text{(No introduction for 0)} \quad \frac{\Gamma \vdash e : 0}{\Gamma \vdash \text{abort } e : T} 0\text{E}
\end{array}$$

Operational semantics

$$\begin{array}{c}
\text{(No rule for unit)} \quad \frac{e_1 \rightsquigarrow e'_1}{\langle e_1, e_2 \rangle \rightsquigarrow \langle e'_1, e_2 \rangle} \text{PAIR1} \quad \frac{e_2 \rightsquigarrow e'_2}{\langle v, e_2 \rangle \rightsquigarrow \langle v, e'_2 \rangle} \text{PAIR2} \\
\\
\frac{}{\text{fst } \langle v_1, v_2 \rangle \rightsquigarrow v_1} \text{PROJ1} \quad \frac{}{\text{snd } \langle v_1, v_2 \rangle \rightsquigarrow v_2} \text{PROJ2} \quad \frac{e \rightsquigarrow e'}{\text{fst } e \rightsquigarrow \text{fst } e'} \text{PROJ3} \quad \frac{e \rightsquigarrow e'}{\text{snd } e \rightsquigarrow \text{snd } e'} \text{PROJ4} \\
\\
\frac{e \rightsquigarrow e'}{\text{L } e \rightsquigarrow \text{L } e'} \text{SUM1} \quad \frac{e \rightsquigarrow e'}{\text{R } e \rightsquigarrow \text{R } e'} \text{SUM2} \\
\\
\frac{e \rightsquigarrow e'}{\text{case}(e, \text{L } x \rightarrow e_1, \text{R } y \rightarrow e_2) \rightsquigarrow \text{case}(e', \text{L } x \rightarrow e_1, \text{R } y \rightarrow e_2)} \text{CASE1} \\
\\
\frac{}{\text{case}(\text{L } v, \text{L } x \rightarrow e_1, \text{R } y \rightarrow e_2) \rightsquigarrow [v/x]e_1} \text{CASE2} \quad \frac{}{\text{case}(\text{R } v, \text{L } x \rightarrow e_1, \text{R } y \rightarrow e_2) \rightsquigarrow [v/y]e_2} \text{CASE3} \\
\\
\frac{e_1 \rightsquigarrow e'_1}{e_1 e_2 \rightsquigarrow e'_1 e_2} \text{APP1} \quad \frac{e_2 \rightsquigarrow e'_2}{v e_2 \rightsquigarrow v e'_2} \text{APP2} \quad \frac{}{(\lambda x : T. e) v \rightsquigarrow [v/x]e} \text{FN} \\
\\
\frac{e \rightsquigarrow e'}{\text{abort } e \rightsquigarrow \text{abort } e'} \text{ABORT}
\end{array}$$

2 Polymorphic λ -Calculus (System F)

Syntax

Types	$T ::= \alpha \mid T_1 \rightarrow T_2 \mid \forall \alpha. T \mid \exists \alpha. T$
Terms	$e ::= x \mid \lambda x : T. e \mid e_1 e_2 \mid \Lambda \alpha. e \mid e T \mid \text{pack}_{\alpha_{\text{abs}}. T_{\text{sig}}}(T_{\text{conc}}, e_{\text{impl}}) \mid \text{let pack}(\alpha, x) = e_{\text{impl}} \text{ in } e_{\text{use}}$
Values	$v ::= \lambda x : T. e \mid \Lambda \alpha. e \mid \text{pack}_{\alpha_{\text{abs}}. T_{\text{sig}}}(T_{\text{conc}}, v_{\text{impl}})$
Type Contexts	$\Theta ::= \cdot \mid \Theta, \alpha$
Term Contexts	$\Gamma ::= \cdot \mid \Gamma, x : T$

Well-formedness of types

$$\frac{\alpha \in \Theta}{\Theta \vdash \alpha \text{ type}} \quad \frac{\Theta \vdash T_1 \text{ type} \quad \Theta \vdash T_2 \text{ type}}{\Theta \vdash T_1 \rightarrow T_2 \text{ type}} \quad \frac{\Theta, \alpha \vdash T \text{ type}}{\Theta \vdash \forall \alpha. T \text{ type}}$$

Well-formedness of term contexts

$$\frac{}{\Theta \vdash \cdot \text{ ctx}} \quad \frac{\Theta \vdash \Gamma \text{ ctx} \quad \Theta \vdash T \text{ type}}{\Theta \vdash \Gamma, x : T \text{ ctx}}$$

Typing rules

$$\begin{array}{c} \frac{x : T \in \Gamma}{\Theta; \Gamma \vdash x : T} \text{HYP} \quad \frac{\Theta \vdash T \text{ type} \quad \Theta; \Gamma, x : T \vdash e : T'}{\Theta; \Gamma \vdash \lambda x : T. e : T \rightarrow T'} \rightarrow I \\[10pt] \frac{\Theta; \Gamma \vdash e_1 : T \rightarrow T' \quad \Theta; \Gamma \vdash e_2 : T}{\Theta; \Gamma \vdash e_1 e_2 : T'} \rightarrow E \\[10pt] \frac{\Theta, \alpha; \Gamma \vdash e : T}{\Theta; \Gamma \vdash \Lambda \alpha. e : \forall \alpha. T} \forall I \quad \frac{\Theta; \Gamma \vdash e : \forall \alpha. T' \quad \Theta \vdash T \text{ type}}{\Theta; \Gamma \vdash e T : [T/\alpha]T'} \forall E \\[10pt] \frac{\Theta, \alpha_{\text{abs}} \vdash T_{\text{sig}} \text{ type} \quad \Theta \vdash T_{\text{conc}} \text{ type} \quad \Theta; \Gamma \vdash e_{\text{impl}} : [T_{\text{conc}}/\alpha_{\text{abs}}]T_{\text{sig}}}{\Theta; \Gamma \vdash \text{pack}_{\alpha_{\text{abs}}. T_{\text{sig}}}(T_{\text{conc}}, e_{\text{impl}}) : \exists \alpha_{\text{abs}}. T_{\text{sig}}} \exists I \\[10pt] \frac{\Theta; \Gamma \vdash e_{\text{impl}} : \exists \alpha_{\text{abs}}. T_{\text{sig}} \quad \Theta, \alpha; \Gamma, x : [\alpha_{\text{abs}}/\alpha]T_{\text{sig}} \vdash e_{\text{use}} : T_{\text{use}} \quad \Theta \vdash T_{\text{use}} \text{ type}}{\Theta; \Gamma \vdash \text{let pack}(\alpha, x) = e_{\text{impl}} \text{ in } e_{\text{use}} : T_{\text{use}}} \exists E \end{array}$$

Operational semantics

(CONG: congruence rule, EVAL: evaluation rule)

$$\begin{array}{c} \frac{e_1 \rightsquigarrow e'_1}{e_1 e_2 \rightsquigarrow e'_1 e_2} \text{CONGFUN} \quad \frac{e_2 \rightsquigarrow e'_2}{v e_2 \rightsquigarrow v e'_2} \text{CONGFUNARG} \quad \frac{}{(\lambda x : T. e) v \rightsquigarrow [v/x]e} \text{FUN EVAL} \\[10pt] \frac{e \rightsquigarrow e'}{e T \rightsquigarrow e' T} \text{CONGFORALL} \quad \frac{}{(\Lambda \alpha. e) T \rightsquigarrow [T/\alpha]e} \text{FORALL EVAL} \\[10pt] \frac{e_{\text{impl}} \rightsquigarrow e'_{\text{impl}}}{\text{pack}_{\alpha_{\text{abs}}. T_{\text{sig}}}(T_{\text{conc}}, e_{\text{impl}}) \rightsquigarrow \text{pack}_{\alpha_{\text{abs}}. T_{\text{sig}}}(T_{\text{conc}}, e'_{\text{impl}})} \text{CONG EXISTS} \\[10pt] \frac{e_{\text{impl}} \rightsquigarrow e'_{\text{impl}}}{\text{let pack}(\alpha, x) = e_{\text{impl}} \text{ in } e_{\text{use}} \rightsquigarrow \text{let pack}(\alpha, x) = e'_{\text{impl}} \text{ in } e_{\text{use}}} \text{CONG EXISTS UNPACK} \\[10pt] \frac{}{\text{let pack}(\alpha, x) = \text{pack}_{\alpha_{\text{abs}}. T_{\text{sig}}}(T_{\text{conc}}, v_{\text{impl}}) \text{ in } e_{\text{use}} \rightsquigarrow [T_{\text{conc}}/\alpha, v_{\text{impl}}/x]e_{\text{use}}} \text{EXISTS EVAL} \end{array}$$

Church encodings

Pairs

$$\begin{aligned}T_1 \times T_2 &\triangleq \forall \alpha. (T_1 \rightarrow T_2 \rightarrow \alpha) \rightarrow \alpha \\ \langle e_1, e_2 \rangle &\triangleq \Lambda \alpha. \lambda k : T_1 \rightarrow T_2 \rightarrow \alpha. k \ e \ e' \\ \text{fst } e &\triangleq e \ T_1 \ (\lambda x : T_1. \lambda y : T_2. x) \\ \text{snd } e &\triangleq e \ T_2 \ (\lambda x : T_1. \lambda y : T_2. y)\end{aligned}$$

Sums

$$\begin{aligned}T_1 + T_2 &\triangleq \forall \alpha. (T_1 \rightarrow \alpha) \rightarrow (T_2 \rightarrow \alpha) \rightarrow \alpha \\ \text{L } e &\triangleq \Lambda \alpha. \lambda f : T_1 \rightarrow \alpha. \lambda g : T_2 \rightarrow \alpha. f \ e \\ \text{R } e &\triangleq \Lambda \alpha. \lambda f : T_1 \rightarrow \alpha. \lambda g : T_2 \rightarrow \alpha. g \ e \\ \text{case}(e, \text{L } x \rightarrow e_1, \text{R } y \rightarrow e_2) : T &\triangleq e \ T \ (\lambda x : T_1 \rightarrow T. e_1) \ (\lambda y : T_2 \rightarrow T. e_2)\end{aligned}$$

Existential types

$$\begin{aligned}\exists \alpha. T_{\text{sig}} &\triangleq \forall \beta. (\forall \alpha. T_{\text{sig}} \rightarrow \beta) \rightarrow \beta \\ \text{pack}_{\alpha_{\text{abs}}, T_{\text{sig}}}(T_{\text{conc}}, e_{\text{impl}}) &\triangleq \Lambda \beta. \lambda k : \forall \alpha_{\text{abs}}. T_{\text{sig}} \rightarrow \beta. k \ T_{\text{conc}} \ e_{\text{impl}} \\ \text{let pack}(\alpha, x) = e_{\text{impl}} \text{ in } e_{\text{use}} : T_{\text{use}} &\triangleq e_{\text{impl}} \ T_{\text{use}} \ (\Lambda \alpha. \lambda x : T_{\text{sig}}. e_{\text{use}})\end{aligned}$$

Booleans

$$\begin{aligned}\text{bool} &\triangleq \forall \alpha. \alpha \rightarrow \alpha \rightarrow \alpha \\ \text{True} &\triangleq \Lambda \alpha. \lambda x : \alpha. \lambda y : \alpha. x \\ \text{False} &\triangleq \Lambda \alpha. \lambda x : \alpha. \lambda y : \alpha. y \\ \text{if } e \text{ then } e_1 \text{ else } e_2 : T &\triangleq e \ T \ e_1 \ e_2\end{aligned}$$

Natural numbers

$$\begin{aligned}\mathbb{N} &\triangleq \forall \alpha. \alpha \rightarrow (\alpha \rightarrow \alpha) \rightarrow \alpha \\ \text{zero} &\triangleq \Lambda \alpha. \lambda z : \alpha. \lambda s : \alpha \rightarrow \alpha. z \\ \text{succ}(e) &\triangleq \Lambda \alpha. \lambda z : \alpha. \lambda s : \alpha \rightarrow \alpha. s \ (e \ \alpha \ z \ s) \\ \text{iter}(e, \text{zero} \rightarrow e_z, \text{succ}(x) \rightarrow e_s) : T &\triangleq e \ T \ e_z \ (\lambda x : T. e_s)\end{aligned}$$

Lists

$$\begin{aligned}\text{list } T &\triangleq \forall \alpha. \alpha \rightarrow (T \rightarrow \alpha \rightarrow \alpha) \rightarrow \alpha \\ [] &\triangleq \Lambda \alpha. \lambda n : \alpha. \lambda c : T \rightarrow \alpha \rightarrow \alpha. n \\ e :: e' &\triangleq \Lambda \alpha. \lambda n : \alpha. \lambda c : T \rightarrow \alpha \rightarrow \alpha. c \ e \ (e' \ \alpha \ n \ c) \\ \text{fold}(e, [] \rightarrow e_n, x :: r \rightarrow e_c) : T' &\triangleq e \ T' \ e_n \ (\lambda x : T. \lambda r : T'. e_c)\end{aligned}$$

3 Monadic λ -Calculus for State

Syntax

Types	$T ::= 1 \mid \mathbb{N} \mid T_1 \rightarrow T_2 \mid \text{ref } T \mid M \ T$
Pure Terms	$e ::= \langle \rangle \mid n \mid \lambda x : T. e \mid e_1 \ e_2 \mid l \mid \{t\}$
Impure Terms	$t ::= \text{new } e \mid !e \mid e := e' \mid \text{let } x = e; t \mid \text{return } e$
Values	$v ::= \langle \rangle \mid n \mid \lambda x : T. e \mid l \mid \{t\}$
Stores	$\sigma ::= \cdot \mid \sigma, l : v$
Contexts	$\Gamma ::= \cdot \mid \Gamma, x : T$
Store Typings	$\Sigma ::= \cdot \mid \Sigma, l : T$

Typing rules

Pure terms

$$\begin{array}{c}
\frac{x : T \in \Gamma}{\Sigma; \Gamma \vdash x : T} \text{HYP} \quad \frac{}{\Sigma; \Gamma \vdash \langle \rangle : 1} \text{1I} \quad \frac{}{\Sigma; \Gamma \vdash n : \mathbb{N}} \text{NI} \\
\\
\frac{\Sigma; \Gamma, x : T \vdash e : T'}{\Sigma; \Gamma \vdash \lambda x : T. e : T \rightarrow T'} \rightarrow \text{I} \quad \frac{\Sigma; \Gamma \vdash e_1 : T \rightarrow T' \quad \Sigma; \Gamma \vdash e_2 : T}{\Sigma; \Gamma \vdash e_1 \ e_2 : T'} \rightarrow \text{E} \\
\\
\frac{l : T \in \Sigma}{\Sigma; \Gamma \vdash l : \text{ref } T} \text{REFBAR} \quad \frac{\Sigma; \Gamma \vdash t \div T}{\Sigma; \Gamma \vdash \{t\} : M \ T} \text{MI}
\end{array}$$

Impure terms

$$\begin{array}{c}
\frac{\Sigma; \Gamma \vdash e : T}{\Sigma; \Gamma \vdash \text{new } e \div \text{ref } T} \text{REFI} \quad \frac{\Sigma; \Gamma \vdash e : \text{ref } T}{\Sigma; \Gamma \vdash !e \div T} \text{REFGET} \quad \frac{\Sigma; \Gamma \vdash e : \text{ref } T \quad \Sigma; \Gamma \vdash e' : T}{\Sigma; \Gamma \vdash e := e' \div 1} \text{REFSET} \\
\\
\frac{\Sigma; \Gamma \vdash e : T}{\Sigma; \Gamma \vdash \text{return } e \div T} \text{MRET} \quad \frac{\Sigma; \Gamma \vdash e : M \ T \quad \Sigma; \Gamma, x : T \vdash t \div T'}{\Sigma; \Gamma \vdash \text{let } x = e; t \div T'} \text{MLET}
\end{array}$$

Store and configuration

$$\frac{}{\Sigma \vdash \cdot : \cdot} \text{STORENIL} \quad \frac{\Sigma \vdash \sigma' : \Sigma' \quad \Sigma; \cdot \vdash v : T}{\Sigma \vdash (\sigma', l : v) : (\Sigma', l : T)} \text{STORECONS} \quad \frac{\Sigma \vdash \sigma : \Sigma \quad \Sigma; \cdot \vdash t \div T}{\langle \sigma; t \rangle : \langle \Sigma; T \rangle} \text{CONFIGOK}$$

Operational semantics

Pure terms

$$\frac{e_1 \rightsquigarrow e'_1}{e_1 \ e_2 \rightsquigarrow e'_1 \ e_2} \quad \frac{e_2 \rightsquigarrow e'_2}{v \ e_2 \rightsquigarrow v \ e'_2} \quad \frac{}{(\lambda x : T. e) \ v \rightsquigarrow [v/x]e}$$

Impure terms

$$\begin{array}{c}
\frac{e \rightsquigarrow e'}{\langle \sigma; \text{new } e \rangle \rightsquigarrow \langle \sigma; \text{new } e' \rangle} \quad \frac{l \notin \text{dom}(\sigma)}{\langle \sigma; \text{new } v \rangle \rightsquigarrow \langle (\sigma, l : v); \text{return } l \rangle} \\
\\
\frac{e \rightsquigarrow e'}{\langle \sigma; !e \rangle \rightsquigarrow \langle \sigma; !e' \rangle} \quad \frac{l : v \in \text{dom}(\sigma)}{\langle \sigma; !l \rangle \rightsquigarrow \langle \sigma; \text{return } v \rangle} \\
\\
\frac{e_1 \rightsquigarrow e'_1}{\langle \sigma; e_1 := e_2 \rangle \rightsquigarrow \langle \sigma; e'_1 := e_2 \rangle} \quad \frac{e_2 \rightsquigarrow e'_2}{\langle \sigma; v := e_2 \rangle \rightsquigarrow \langle \sigma; v := e'_2 \rangle}
\end{array}$$

$$\begin{array}{c}
\frac{}{\langle(\sigma, l : v, \sigma'); l := v\rangle \rightsquigarrow \langle(\sigma, l : v', \sigma'); \text{return } \langle\rangle\rangle} \\
\\
\frac{e \rightsquigarrow e'}{\langle\sigma; \text{return } e\rangle \rightsquigarrow \langle\sigma; \text{return } e'\rangle} \qquad \frac{e \rightsquigarrow e'}{\langle\sigma; \text{let } x = e; t\rangle \rightsquigarrow \langle\sigma; \text{let } x = e'; t\rangle} \\
\\
\frac{}{\langle\sigma; \text{let } x = \{\text{return } v\}; t\rangle \rightsquigarrow \langle\sigma; [v/x]t\rangle} \qquad \frac{\langle\sigma; t_1\rangle \rightsquigarrow \langle\sigma'; t'_1\rangle}{\langle\sigma; \text{let } x = \{t_1\}; t_2\rangle \rightsquigarrow \langle\sigma'; \text{let } x = \{t'_1\}; t_2\rangle}
\end{array}$$

4 Monadic λ -Calculus for I/O

Syntax

Types	$T ::= 1 \mid \mathbb{N} \mid T_1 \rightarrow T_2 \mid M_{\text{IO}} T$
Pure Terms	$e ::= \langle\rangle \mid n \mid \lambda x : T. e \mid e_1 e_2 \mid l \mid \{t\}$
Impure Terms	$t ::= \text{print } e \mid \text{let } x = e; t \mid \text{return } e$
Values	$v ::= \langle\rangle \mid n \mid \lambda x : T. e \mid \{t\}$
Output Tokens	$\omega ::= \cdot \mid n :: \omega$
Contexts	$\Gamma ::= \cdot \mid \Gamma, x : T$

Typing rules

Pure terms

$$\begin{array}{c}
\frac{x : T \in \Gamma}{\Gamma \vdash x : T} \text{HYP} \qquad \frac{}{\Gamma \vdash \langle\rangle : 1} \text{1I} \qquad \frac{}{\Gamma \vdash n : \mathbb{N}} \text{NI} \\
\\
\frac{\Gamma, x : T \vdash e : T'}{\Gamma \vdash \lambda x : T. e : T \rightarrow T'} \rightarrow\text{I} \qquad \frac{\Gamma \vdash e_1 : T \rightarrow T' \quad \Gamma \vdash e_2 : T}{\Gamma \vdash e_1 e_2 : T'} \rightarrow\text{E} \qquad \frac{\Gamma \vdash t \div T}{\Gamma \vdash \{t\} : M_{\text{IO}} T} \text{MI}
\end{array}$$

Impure terms

$$\frac{\Gamma \vdash e : \mathbb{N}}{\Gamma \vdash \text{print } e \div 1} \text{MPRINT} \qquad \frac{\Gamma \vdash e : T}{\Gamma \vdash \text{return } e \div T} \text{MRET} \qquad \frac{\Gamma \vdash e : M_{\text{IO}} T \quad \Gamma, x : T \vdash t \div T'}{\Gamma \vdash \text{let } x = e; t \div T'} \text{MLET}$$

Operational semantics

Pure terms

$$\frac{e_1 \rightsquigarrow e'_1}{e_1 e_2 \rightsquigarrow e'_1 e_2} \qquad \frac{e_2 \rightsquigarrow e'_2}{v e_2 \rightsquigarrow v e'_2} \qquad \frac{}{(\lambda x : T. e) v \rightsquigarrow [v/x]e}$$

Impure terms

$$\begin{array}{c}
\frac{e \rightsquigarrow e'}{\langle\omega; \text{print } e\rangle \rightsquigarrow \langle\omega; \text{print } e'\rangle} \qquad \frac{}{\langle\omega; \text{print } n\rangle \rightsquigarrow \langle(n :: \omega); \text{return } \langle\rangle\rangle} \\
\\
\frac{e \rightsquigarrow e'}{\langle\omega; \text{return } e\rangle \rightsquigarrow \langle\omega; \text{return } e'\rangle} \qquad \frac{e \rightsquigarrow e'}{\langle\omega; \text{let } x = e; t\rangle \rightsquigarrow \langle\omega; \text{let } x = e'; t\rangle} \\
\\
\frac{}{\langle\omega; \text{let } x = \{\text{return } v\}; t\rangle \rightsquigarrow \langle\omega; [v/x]t\rangle} \qquad \frac{\langle\omega; t_1\rangle \rightsquigarrow \langle\omega'; t'_1\rangle}{\langle\omega; \text{let } x = \{t_1\}; t_2\rangle \rightsquigarrow \langle\omega'; \text{let } x = \{t'_1\}; t_2\rangle}
\end{array}$$