Programming Language Theory

CSCI-740  
Term 20191  
Handout 2  
October 11, 2019

Type Safety of STLC (with constants)

Syntax

\[
\begin{align*}
    e &::= c \mid x \mid \lambda x. e \mid e \ e \\
v &::= c \mid \lambda x. e
\end{align*}
\]

Work with terms “up to renaming of bound variables” (“up to alpha-conversion”).

Substitution

\[
\begin{align*}
    FV(c) &= \{\}\ \\
    FV(x) &= \{x\} \\
    FV(e_\ f \ e_\ a) &= FV(e_\ f) \cup FV(e_\ a) \\
    FV(\lambda x. \ e_\ b) &= FV(e_\ b) \setminus \{x\}
\end{align*}
\]

\[
e_1[e_2/z] = e_3
\]

\[
\begin{array}{c}
\frac{e[z/x]}{e[z/x] = e} & \quad \frac{x = z}{x[z/x] = e} & \quad \frac{x \neq z}{x[z/x] = x} \\
\hline
\frac{e_\ b[z/x] = e_\ b'}{e_\ b[z/x] = \lambda x. e_\ b'} & \quad \frac{x \neq z \land x \notin FV(e)}{x[z/x] = e_\ b'} & \quad \frac{e_\ f[z/x] = e_\ f'}{e_\ a[z/x] = e_\ a'}
\end{array}
\]

Substitution usually treated as a metafunction, not a judgement.

Operational Semantics

Small-step, left-to-right, call-by-value (CBV) operational semantics:

\[
\begin{array}{c}
\frac{e \rightarrow_{cbv} e'}{e \rightarrow e'} \\
\hline
E\text{-Apply} & E\text{-AppF} & E\text{-AppA}
\end{array}
\]

\[
\begin{align*}
    (\lambda x. e_\ b) v_\ a &\rightarrow_{cbv} e_\ b[v_\ a/x] \\
    e_\ f e_\ a &\rightarrow_{cbv} e_\ f' e_\ a \\
    e_\ a &\rightarrow_{cbv} e_\ a'
\end{align*}
\]

We say that an expression \( e \) is stuck if \( e \) is not a value, and there is no \( e' \) such that \( e \rightarrow_{cbv} e' \)

We say that an expression \( e \) gets stuck if \( e \rightarrow_{cbv}^* e' \), and \( e' \) is stuck.
Type System

Type system to classify (accept or reject) λ-terms.

\[ \tau ::= \text{int} \mid \tau \rightarrow \tau \]
\[ \Gamma ::= \cdot \mid \Gamma, x : \tau \]

C-Hit
\[ \Gamma, x : \tau_x @ x \leadsto \tau_x \]

C-Miss
\[ x \neq y \quad \Gamma' @ x \leadsto \tau \]
\[ \Gamma', y : \tau_y @ x \leadsto \tau \]

Γ ⊢ e : τ

T-Const
\[ \Gamma \vdash c : \text{int} \]

T-Var
\[ \Gamma @ \vdash x : \tau \]
\[ \Gamma \vdash x : \tau \]

T-Lam
\[ \Gamma, x : \tau_a \vdash e_b : \tau_r \]
\[ \Gamma \vdash \lambda x. e_b : \tau_a \rightarrow \tau_r \]

T-App
\[ \Gamma \vdash e_f : \tau_a \rightarrow \tau_r \quad \Gamma \vdash e_a : \tau_a \]
\[ \Gamma \vdash e_f e_a : \tau_r \]

Type Safety Theorems/Lemmas

Theorem (Type Safety):
If \( \cdot \vdash e : \tau \) and \( e \rightarrow^*_{\text{cbv}} e' \),
then either \( e' \) is a value or there exists \( e'' \) such that \( e' \rightarrow_{\text{cbv}} e'' \).

- **Lemma (Progress):**
  If \( \cdot \vdash e' : \tau \), then either \( e' \) is a value or there exists an \( e'' \) such that \( e \rightarrow_{\text{cbv}} e'' \).

  - **Lemma (Canonical Forms; int):**
    If \( \cdot \vdash v : \text{int} \), then \( v = c \) (for some \( c \)).

  - **Lemma (Canonical Forms; \( \tau_a \rightarrow \tau_r \)):**
    If \( \cdot \vdash v : \tau_a \rightarrow \tau_r \), then \( v = \lambda x. e_b \) (for some \( \lambda x. e_b \)).

- **Lemma (Preservation):**
  If \( \cdot \vdash e : \tau \) and \( e \rightarrow_{\text{cbv}} e' \), then \( \cdot \vdash e' : \tau \).

- **Lemma (Substitution):**
  If \( \Gamma, z : \tau_z \vdash e_1 : \tau \) and \( \Gamma \vdash e_2 : \tau_z \), then \( \Gamma \vdash e_1[e_2/z] : \tau \).

  * **Lemma (Exchange):**
    If \( \Gamma, x : \tau_x, y : \tau_y \vdash e : \tau \) and \( x \neq y \), then \( \Gamma, y : \tau_y, x : \tau_x \vdash e : \tau \).

  * **Lemma (Weakening):**
    If \( \Gamma \vdash e : \tau \) and \( x \notin \text{Dom}(\Gamma) \), then \( \Gamma, x : \tau_x \vdash e : \tau \).
Type Safety Proof

A program that type checks does not get stuck.

Theorem (Type Safety):
If \( \cdot \vdash e : \tau \) and \( e \rightarrow^*_{cbv} e' \),
then either \( e' \) is a value or there exists \( e'' \) such that \( e' \rightarrow_{cbv} e'' \).

Comments: The Type Safety Theorem follows as a simple corollary to the Progress and Preservation Lemmas stated and proven below.

Proof (assuming Preservation and Progress):
By structural induction on (the derivation of) \( e \rightarrow^*_{cbv} e' \).

- \( e \rightarrow^*_{cbv} e' \equiv e \rightarrow^*_{cbv} e \) :
  Therefore, \( e' = e \).
  We must show that either \( e' \) is a value or there exists \( e'' \) such that \( e' \rightarrow^*_{cbv} e'' \).
  From \( \cdot \vdash e : \tau \) and \( e = e' \), we have \( \cdot \vdash e' : \tau \).
  By Progress applied to \( \cdot \vdash e' : \tau \), we have either \( e' \) is a value or there exists \( e'' \) such that \( e' \rightarrow_{cbv} e'' \).

- \( e \rightarrow^*_{cbv} e' \equiv e \rightarrow^*_{cbv} e' \) :
  Therefore, we have \( e \rightarrow_{cbv} e' \) and \( e' \rightarrow^*_{cbv} e'' \).
  We must show that either \( e' \) is a value or there exists \( e'' \) such that \( e' \rightarrow^*_{cbv} e'' \).
  By Preservation applied to \( \cdot \vdash e : \tau \) and \( e \rightarrow^*_{cbv} e' \), we have \( \cdot \vdash e' : \tau \).
  By the induction hypothesis applied to \( e' \rightarrow^*_{cbv} e' \) with \( \cdot \vdash e' : \tau \),
  we have either \( e' \) is a value or there exists \( e'' \) such that \( e' \rightarrow_{cbv} e'' \).
**Lemma (Progress):** If \( \cdot \vdash e : \tau \), then either \( e \) is a value or there exists an \( e' \) such that \( e \rightarrow_{cbv} e' \).

**Proof (assuming Canonical Forms):**

By induction on (the derivation of) \( \cdot \vdash e : \tau \):

- **T-Const** concludes the derivation of \( \cdot \vdash e : \tau \):
  
  Therefore, \( e = c \) and \( \tau = \text{int} \).
  
  We must show that either \( e \) is a value or there exists an \( e' \) such that \( e \rightarrow_{cbv} e' \).
  
  We have \( e = c \) is a value.

- **T-Var** concludes the derivation of \( \cdot \vdash e : \tau \):
  
  Therefore, \( \cdot \vdash x : \tau_0 \vdash e_0 : \tau_r \), \( e = \lambda x. e_0 \), and \( \tau = \tau_0 \rightarrow \tau_r \).
  
  We must show that either \( e \) is a value or there exists an \( e' \) such that \( e \rightarrow_{cbv} e' \).
  
  We have \( e = \lambda x. e_0 \) is a value.

- **T-Lam** concludes the derivation of \( \cdot \vdash e : \tau \):
  
  Therefore, \( \cdot \vdash \lambda x. e : \tau_0 \rightarrow \tau_r \), \( \cdot \vdash e : \tau_0 \vdash e : \tau_r \), and \( \tau = \tau_0 \rightarrow \tau_r \).
  
  We must show that either \( e \) is a value or there exists an \( e' \) such that \( e \rightarrow_{cbv} e' \).
  
  By the induction hypothesis applied to \( \cdot \vdash e : \tau_0 \rightarrow \tau_r \), we have either

  - \( e \) is a value:
    
    Therefore, \( e \) = \( v_f \).
    
    By the induction hypothesis applied to \( \cdot \vdash e_0 : \tau_0 \), we have either
    
    * \( e_0 \) is a value:
      
      Therefore, \( e_0 = v_a \).
      
      From \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r \), \( e_f = v_f \), we have \( \cdot \vdash v_f : \tau_a \rightarrow \tau_r \).
      
      By *Canonical Forms* applied to \( \cdot \vdash v_f : \tau_a \rightarrow \tau_r \), we have \( v_f = \lambda x. e_b \).
      
      From E-Apply, we can construct the derivation \( (\lambda x. e_b) v_a \rightarrow_{cbv} e_b[v_a/x] \); therefore, we have \( (\lambda x. e_b) v_a \rightarrow_{cbv} e_b[v_a/x] \).
      
      Take \( e' = e_b[v_a/x] \).
      
      From \( (\lambda x. e_b) v_a \rightarrow_{cbv} e_b[v_a/x] \), \( e = e_f e_a \), \( e_f = v_f \), \( e_a = v_a \), \( v_f = \lambda x. e_b \), and \( e' = e_b[v_a/x] \), we have \( e \rightarrow_{cbv} e' \).
    
    * there exists an \( e'_a \) such that \( e \rightarrow_{cbv} e'_a \):
      
      From E-Apply and \( e_a \rightarrow_{cbv} e'_a \), we can construct the derivation \( e_a \rightarrow_{cbv} e'_a \); therefore, we have \( v_f e_a \rightarrow_{cbv} v_f e'_a \).
      
      Take \( e' = v_f e'_a \).
      
      From \( v_f e_a \rightarrow_{cbv} v_f e'_a \), \( e = e_f e_a \), \( v_f = e_f \), and \( e' = v_f e'_a \), we have \( e \rightarrow_{cbv} e' \).
  
  - there exists an \( e'_f \) such that \( e \rightarrow_{cbv} e'_f \):
     
     From E-Apply and \( e_f \rightarrow_{cbv} e'_f \), we can construct the derivation \( e_f \rightarrow_{cbv} e'_f \); therefore, we have \( e_f e_a \rightarrow_{cbv} e'_f e_a \).
     
     Take \( e' = e'_f e_a \).
     
     From \( e_f e_a \rightarrow_{cbv} e'_f e_a \), \( e = e_f e_a \), and \( e' = e'_f e_a \), we have \( e \rightarrow_{cbv} e' \).
Lemma (Canonical Forms): If $\cdot \vdash v : \tau$, then

1. if $\tau = \text{int}$, then $v = c$ (for some $c$)
2. if $\tau = \tau_a \to \tau_r$, then $v = \lambda x. e_b$ (for some $\lambda x. e_b$)

Proof:
(By inspection of the typing rules.)

1. $\tau = \text{int}$:
   By assumption, $\cdot \vdash v : \text{int}$.
   Only T-CONST can derive $\cdot \vdash v : \text{int}$; therefore, $v = c$ (for some $c$).

2. $\tau = \tau_a \to \tau_r$:
   By assumption, $\cdot \vdash v : \tau_a \to \tau_r$.
   Only T-LAM can derive $\cdot \vdash v : \tau_a \to \tau_r$; therefore, $v = \lambda x. e_b$ (for some $\lambda x. e_b$).
Lemma (Preservation): If \( \vdash e : \tau \) and \( e \rightarrow_{\text{cbv}} e' \), then \( \vdash e' : \tau \).

Proof (assuming Substitution):

By induction on (the derivation of) \( \vdash e : \tau \):

- **T-Const** concludes the derivation of \( \vdash e : \tau \):
  Therefore, \( e = c \) and \( \tau = \text{int} \).
  From \( e \rightarrow_{\text{cbv}} e' \) and \( e = c \), we have \( c \rightarrow_{\text{cbv}} e' \).
  But \( c \rightarrow_{\text{cbv}} e' \) is contradictory; there is no derivation of such a judgement.
  Therefore, vacuously, \( \vdash e' : \tau \).

- **T-VAR** concludes the derivation of \( \vdash e : \tau \):
  Therefore, \( e = x \) and \( \overline{\emptyset} x \sim \tau \).
  From \( e \rightarrow_{\text{cbv}} e' \) and \( e = x \), we have \( x \rightarrow_{\text{cbv}} e' \).
  But \( x \rightarrow_{\text{cbv}} e' \) is contradictory; there is no derivation of such a judgement.
  Therefore, vacuously, \( \vdash e' : \tau \).

- **T-LAM** concludes the derivation of \( \vdash e : \tau \):
  Therefore, \( \lambda x. e \rightarrow_{\text{cbv}} e' \) and \( e = \lambda x. e_b \), and \( \tau = \tau_a \rightarrow \tau_r \).
  From \( e \rightarrow_{\text{cbv}} e' \) and \( e = \lambda x. e_b \), we have \( \lambda x. e_b \rightarrow_{\text{cbv}} e' \).
  But \( \lambda x. e_b \rightarrow_{\text{cbv}} e' \) is contradictory; there is no derivation of such a judgement.
  Therefore, vacuously, \( \vdash e' : \tau \).

- **T-APP** concludes the derivation of \( \vdash e : \tau \):
  Therefore, \( \vdash e_f : \tau_a \rightarrow \tau_r \), \( \vdash e_a : \tau_a \), \( e = e_f e_a \), and \( \tau = \tau_r \).
  From \( e \rightarrow_{\text{cbv}} e' \) and \( e = e_f e_a \), we have \( e_f e_a \rightarrow_{\text{cbv}} e' \).
  By cases on (the derivation of) \( e_f e_a \rightarrow_{\text{cbv}} e' \):
    - **E-APPLY** concludes the derivation of \( e_f e_a \rightarrow_{\text{cbv}} e' \):
      Therefore, \( e_f = \lambda x. e_b \), \( e_a = v_a \), and \( e' = e_b[v_a/x] \).
      From \( e' = e_b[v_a/x] \) and \( e_a = v_a \), we have \( e' = e_b[v_a/x] \).
      From \( \vdash e_f : \tau_a \rightarrow \tau_r \) and \( e_f = \lambda x. e_b \), we have \( \vdash \lambda x. e_b : \tau_a \rightarrow \tau_r \).
      By inversion of \( \vdash \lambda x. e_b : \tau_a \rightarrow \tau_r \), we have \( \vdash x : \tau_a \vdash e_b : \tau_r \).
      By **Substitution** applied to \( x : \tau_a \vdash e_b : \tau_r \) and \( \vdash e_a : \tau_a \), we have \( \vdash e_b[e_a/x] : \tau_r \).
      From \( \vdash e_b[e_a/x] : \tau_r \), \( e' = e_b[e_a/x] \), and \( \tau = \tau_r \), we have \( \vdash e' : \tau \).
    - **E-APPF** concludes the derivation of \( e_f e_a \rightarrow_{\text{cbv}} e' \):
      Therefore, \( e_f \rightarrow_{\text{cbv}} e'_f \) and \( e' = e'_f e_a \).
      By the induction hypothesis applied to \( \vdash e_f : \tau_a \rightarrow \tau_r \) and \( e_f \rightarrow_{\text{cbv}} e'_f \), we have \( \vdash e'_f : \tau_a \rightarrow \tau_r \).
      From **T-APP**, \( \vdash e'_f : \tau_a \rightarrow \tau_r \), and \( \vdash e_a : \tau_a \),
      we can construct the derivation \( \vdash e'_f : \tau_a \rightarrow \tau_r \).
      therefore, we have \( \vdash e'_f e_a : \tau_r \).
      From \( \vdash e'_f e_a : \tau_r \), \( e' = e'_f e_a \), and \( \tau = \tau_r \), we have \( \vdash e' : \tau_r \).
    - **E-APPFA** concludes the derivation of \( e_f e_a \rightarrow_{\text{cbv}} e' \):
      Therefore, \( e_a \rightarrow_{\text{cbv}} e'_a \) and \( e_f = v_f \), and \( e' = v_f e'_a \).
      From \( e' = v_f e'_a \) and \( e_f = v_f \), we have \( e' = e_f e'_a \).
      By the induction hypothesis applied to \( e_a \rightarrow_{\text{cbv}} e'_a \) and \( \vdash e_a : \tau_a \), we have \( \vdash e'_a : \tau_a \).
      From **T-APP**, \( \vdash e_f : \tau_a \rightarrow \tau_r \), and \( \vdash e'_a : \tau_a \),
      we can construct the derivation \( \vdash e_f : \tau_a \rightarrow \tau_r \).
      therefore, we have \( \vdash e_f e'_a : \tau_r \).
      From \( \vdash e_f e'_a : \tau_r \), \( e' = e_f e'_a \), and \( \tau = \tau_r \), we have \( \vdash e' : \tau_r \).
Lemma (Substitution): If $\Gamma, z : \tau_z \vdash e_1 : \tau$ and $\Gamma \vdash e_2 : \tau_z$, then $\Gamma \vdash e_1[e_2/z] : \tau$.

Comments: The proof of the Preservation Lemma only requires a Substitution Lemma where $\Gamma = \cdot$. However, proving the Substitution Lemma itself requires the stronger induction hypothesis.

Proof (assuming Exchange and Weakening):
By structural induction on $e_1$.

- $e_1 \equiv c$:
  By assumption, we have $\Gamma, z : \tau_z \vdash e_1 : \tau$ and $\Gamma \vdash e_2 : \tau_z$.
  We must show that $\Gamma \vdash e_1[e_2/z] : \tau$.
  From $\Gamma, z : \tau_z \vdash e_1 : \tau$ and $e_1 = c$, we have $\Gamma, z : \tau_z \vdash c : \tau$.
  By inversion of $\Gamma, z : \tau_z \vdash c : \tau$, we have $\tau = \text{int}$.
  From T-Const, we can construct the derivation $\Gamma \vdash c : \text{int}$;
  therefore, we have $\Gamma \vdash c : \text{int}$.
  By definition of substitution, we have $c[e_2/z] = c$.
  From $\Gamma \vdash c : \text{int}$, $e_1 = c$, $\tau = \text{int}$, and $c[e_2/z] = c$, we have $\Gamma \vdash e_1[e_2/z] : \tau$.

- $e_1 \equiv x$:
  By assumption, we have $\Gamma, z : \tau_z \vdash e_1 : \tau$ and $\Gamma \vdash e_2 : \tau_z$.
  We must show that $\Gamma \vdash e_1[e_2/z] : \tau$.
  From $\Gamma, z : \tau_z \vdash e_1 : \tau$ and $e_1 = x$, we have $\Gamma, z : \tau_z \vdash x : \tau$.
  By inversion of $\Gamma, z : \tau_z \vdash x : \tau$, we have $\Gamma, z : \tau_z @ x \sim \tau$.
  By cases on (the derivation of) $\Gamma, z : \tau_z @ x \sim \tau$.
    - C-Hit concludes the derivation of $\Gamma, z : \tau_z @ x \sim \tau$:
      Therefore, $x = z$ and $\tau = \tau_z$.
      By definition of substitution, we have $z[e_2/z] = e_2$.
      From $\Gamma \vdash e_2 : \tau_z$, $e_1 = x$, $x = z$, $\tau = \tau_z$, and $z[e_2/z] = e_2$,
      we have $\Gamma \vdash e_1[e_2/z] : \tau$.
    - C-Miss concludes the derivation of $\Gamma, z : \tau_z @ x \sim \tau$:
      Therefore, $x \neq z$ and $\Gamma @ x \sim \tau$.
      From T-VAR, we can construct the derivation $\Gamma @ x \sim \tau$;
      therefore, we have $\Gamma \vdash x : \tau$.
      By definition of substitution and $x \neq z$, we have $x[e_2/z] = x$.
      From $\Gamma \vdash x : \tau$, $e_1 = x$, and $x[e_2/z] = x$,
      we have $\Gamma \vdash e_1[e_2/z] : \tau$.

(continued)
• \(e_1 \equiv \lambda x. e_b\):
  By “up to \(\alpha\)-conversion”, we ensure \(x \neq z\) and \(x \notin \text{Dom}(\Gamma)\).
  By assumption, we have \(\Gamma, z : \tau_z \vdash e_1 : \tau\) and \(\Gamma \vdash e_2 : \tau_z\).
  We must show that \(\Gamma \vdash e_1[e_2/z] : \tau\).
  From \(\Gamma, z : \tau_z \vdash e_1 : \tau\) and \(e_1 = \lambda x. e_b\), we have \(\Gamma, z : \tau_z \vdash \lambda x. e_b : \tau\).
  By inversion of \(\Gamma, z : \tau_z \vdash e_2 : \tau\), we have \(\Gamma, z : \tau_z, x : \tau_a \vdash e_b : \tau\) and \(\tau = \tau_a \rightarrow \tau_r\).
  By \text{Exchange} applied to \(\Gamma, z : \tau_z, x : \tau_a \vdash e_b : \tau\) and \(x \neq z\), we have \(\Gamma, x : \tau_a, z : \tau_z \vdash e_b : \tau_r\).
  By \text{Weakening} applied to \(\Gamma \vdash e_2 : \tau_z\) and \(x \notin \text{Dom}(\Gamma)\), we have \(\Gamma, x : \tau_a \vdash e_2 : \tau_z\).
  By the induction hypothesis applied to \(e_b\) with \(\Gamma, x : \tau_a, z : \tau_z \vdash e_b : \tau_r\) and \(\Gamma, x : \tau_a \vdash e_2 : \tau_z\), we have \(\Gamma, x : \tau_a \vdash e_b[e_2/z] : \tau_r\).

  From \text{T-LAM} and \(\Gamma, x : \tau_a \vdash e_b[e_2/z] : \tau_r\), we can construct the derivation \(\Gamma \vdash \lambda x. e_b[e_2/z] : \tau_a \rightarrow \tau_r\).
  Therefore, we have \(\Gamma \vdash \lambda x. e_b[e_2/z] : \tau_a \rightarrow \tau_r\).
  From \(x \notin \text{Dom}(\Gamma)\) and \(\Gamma \vdash e_2 : \tau_z\), we have \(x \notin \text{FV}(e_2)\).
  By definition of substitution and \(x \neq z\) and \(x \notin \text{FV}(e_2)\), we have \((\lambda x. e_b)[e_2/z] = \lambda x. e_b[e_2/z]\).
  From \(\Gamma \vdash \lambda x. e_b[e_2/z] : \tau_a \rightarrow \tau_r, e_1 = \lambda x. e_b, \tau = \tau_a \rightarrow \tau_r\), and \((\lambda x. e_b)[e_2/z] = \lambda x. e_b[e_2/z]\),
  we have \(\Gamma \vdash e_1[e_2/z] : \tau\).

• \(e_1 \equiv ef \ e_a\):
  By assumption, we have \(\Gamma, z : \tau_z \vdash e_1 : \tau\) and \(\Gamma \vdash e_2 : \tau_z\).
  We must show that \(\Gamma \vdash e_1[e_2/z] : \tau\).
  From \(\Gamma, z : \tau_z \vdash e_1 : \tau\) and \(e_1 = ef \ e_a\), we have \(\Gamma, z : \tau_z \vdash ef \ e_a : \tau\).
  By inversion of \(\Gamma, z : \tau_z \vdash e_f \ e_a : \tau\), we have \(\Gamma, z : \tau_z \vdash e_f : \tau_a \rightarrow \tau_r\), \(\Gamma, z : \tau_z \vdash e_a : \tau_a\), and \(\tau = \tau_r\).
  By the induction hypothesis applied to \(e_f\) with \(\Gamma, z : \tau_z \vdash e_f : \tau_a \rightarrow \tau_r\) and \(\Gamma \vdash e_2 : \tau_z\),
  we have \(\Gamma \vdash e_f[e_2/z] : \tau_a \rightarrow \tau_r\).
  By the induction hypothesis applied to \(e_a\) with \(\Gamma, z : \tau_z \vdash e_a : \tau_a\) and \(\Gamma \vdash e_2 : \tau_z\),
  we have \(\Gamma \vdash e_a[e_2/z] : \tau_a\).
  From \text{T-APP}, \(\Gamma \vdash e_f[e_2/z] : \tau_a \rightarrow \tau_r\), and \(\Gamma \vdash e_a[e_2/z] : \tau_a\),
  we can construct the derivation \(\Gamma \vdash e_f[e_2/z] \ e_a[e_2/z] : \tau_r\).
  Therefore, we have \(\Gamma \vdash e_f[e_2/z] \ e_a[e_2/z] : \tau_r\).
  By definition of substitution, we have \((ef \ e_a)[e_2/z] = ef[e_2/z] \ e_a[e_2/z]\).
  From \(\Gamma \vdash ef[e_2/z] \ e_a[e_2/z] : \tau_r, e_1 = ef \ e_a, \tau = \tau_r\), and \((ef \ e_a)[e_2/z] = ef[e_2/z] \ e_a[e_2/z]\),
  we have \(\Gamma \vdash e_1[e_2/z] : \tau\).

**Lemma (Exchange):** If \(\Gamma, x : \tau_x, y : \tau_y \vdash e : \tau\) and \(x \neq y\), then \(\Gamma, y : \tau_y, x : \tau_x \vdash e : \tau\).

*Comments:* The Exchange Lemma is a technical lemma, whose proof is omitted but is not difficult. (The proof is by induction on the structure of \(e\).)

**Lemma (Weakening):** If \(\Gamma \vdash e : \tau\) and \(x \notin \text{Dom}(\Gamma)\), then \(\Gamma, x : \tau_x \vdash e : \tau\).

*Comments:* The Weakening Lemma is a technical lemma, whose proof is omitted but is not difficult. (The proof is by induction on the structure of \(e\).)