Syntax

\[ e ::= c \mid x \mid \lambda x.\ e \mid e\ e \]
\[ v ::= c \mid \lambda x.\ e \]

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

Substitution

\[
\begin{align*}
FV(c) &= \{\} \\
FV(x) &= \{x\} \\
FV(e_f e_a) &= FV(e_f) \cup FV(e_a) \\
FV(\lambda x. e_b) &= FV(e_b) \setminus \{x\}
\end{align*}
\]

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

Substitution usually treated as a metafunction, not a judgement.

Operational Semantics

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

\[
\begin{align*}
E\cdot\text{AppF} & \quad e_f \rightarrow_{\text{cbv}} e_f' \quad e_f e_a \rightarrow_{\text{cbv}} e_f' e_a' \\
E\cdot\text{AppA} & \quad e_a \rightarrow_{\text{cbv}} e_a' \quad e_f e_a \rightarrow_{\text{cbv}} e_f' e_a' \\
E\cdot\text{Apply} & \quad (\lambda x. e_b) v_a \rightarrow_{\text{cbv}} e_b[v_a/x]
\end{align*}
\]

\[ e \rightarrow_{\text{cbv}} e' \]

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

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

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

\[
\begin{align*}
\tau & ::= \text{int} \mid \tau \rightarrow \tau \\
\Gamma & ::= \cdot \mid \Gamma, x : \tau
\end{align*}
\]

\( \Gamma \vdash x : \tau \)

C-Hit

\[
\Gamma, x : \tau \vdash x : \tau
\]

C-Miss

\[
\Gamma \not\vdash x : \tau
\]

\( \Gamma \vdash e : \tau \)

T-Const

\[
\Gamma \vdash c : \text{int}
\]

T-Var

\[
\Gamma \vdash x : \tau
\]

T-Lam

\[
\Gamma, x : \tau \vdash e_b : \tau_r
\]

\[
\Gamma \vdash \lambda x. e_b : \tau_a \rightarrow \tau_r
\]

T-App

\[
\begin{align*}
\Gamma \vdash e_f : \tau_a \rightarrow \tau_r \\
\Gamma \vdash e_a : \tau_a
\end{align*}
\]

\[
\Gamma \vdash e_f e_a : \tau_r
\]

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 \( x \not\in \text{Dom}(\Gamma) \), 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) \( e \rightarrow_{cbv}^* e' \).

- \( e \rightarrow_{cbv}^* e' \equiv e \rightarrow_{cbv} e \) : E\(^*\)-Zero
  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 \( e' \rightarrow_{cbv} e'' \).

- \( e \rightarrow_{cbv}^* e' \equiv e \rightarrow_{cbv} e \rightarrow_{cbv} e' \) : E\(^*\)-Step
  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, \( e = v \) 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 \( e = v \) is a value.

- **T-Var** concludes the derivation of \( \cdot \vdash e : \tau \):
  Therefore, \( e = \lambda x. e_b \) and \( \tau = \tau_a \rightarrow \tau_r \).
  We must show that either \( e \) is a value or there exists an \( e' \) such that \( e \rightarrow_{cbv} e' \).
  But, \( \cdot \vdash e : \tau \) is contradictory; there is no derivation of such a judgement.
  Therefore, vacuously, either \( e \) is a value or there exists an \( e' \) such that \( e \rightarrow_{cbv} e' \).

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

- **T-App** concludes the derivation of \( \cdot \vdash e : \tau \):
  Therefore, \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r \), \( \cdot \vdash e_a : \tau_a = e_f e_a, \text{ and } \tau = \tau_r. \)
  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 \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r \), we have either
  
  - \( e_f \) is a value:
    Therefore, \( e_f = v_f \).
    By the induction hypothesis applied to \( \cdot \vdash e_a : \tau_a \), we have either
      
      * \( e_a \) is a value:
        Therefore, \( e_a = v_a \).
        From \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r \) and \( e_f = v_f \), 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 \( \lambda x. e_b v_a \rightarrow_{cbv} e_b[v_a/x] \), \( e = e_f e_a, e_f = v_f, e_a = v_a, v_f = \lambda x. e_b, \text{ and } e' = e_b[v_a/x] \), we have \( e \rightarrow_{cbv} e' \).
      
      * there exists an \( e'_a \) such that \( e_a \rightarrow_{cbv} e'_a \):
        From E-AppF and \( e_a \rightarrow_{cbv} e'_a \), we can construct the derivation \( e_a \rightarrow_{cbv} e_a ; v_f e_a \rightarrow_{cbv} v_f e'_a \); therefore, we have \( v_f e_a \rightarrow_{cbv} v_f e'_a \).
        Take \( e' = v_f e'_a \).
        From \( v_f e_a \rightarrow_{cbv} v_f e'_a, \text{ and } e = e_f e_a, v_f = e_f, \text{ and } e' = v_f e'_a \), we have \( e \rightarrow_{cbv} e' \).

  - there exists an \( e'_f \) such that \( e_f \rightarrow_{cbv} e'_f \):
    From E-AppF and \( e_f \rightarrow_{cbv} e'_f \), we can construct the derivation \( e_f \rightarrow_{cbv} e'_f ; e_f e_a \rightarrow_{cbv} e_f e_a \); therefore, we have \( e_f e_a \rightarrow_{cbv} e_f e_a \).
    Take \( e' = e'_f e_a \).
    From \( e_f e_a \rightarrow_{cbv} e'_f e_a, e = e_f e_a, \text{ and } e' = e'_f e_a \), we have \( e \rightarrow_{cbv} e' \).
Lemma (Canonical Forms): If $\cdot \vdash v : \tau$, then

1. if $\tau$ = int, then $v = c$ (for some $c$)

2. if $\tau = \tau_a \rightarrow \tau_r$, then $v = \lambda x. e_b$ (for some $\lambda x. e_b$)

Proof:
(By inspection of the typing rules.)

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

Proof (assuming Substitution):

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

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

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

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

- **T-APP** concludes the derivation of \( \cdot \vdash e : \tau \):
  Therefore, \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r, \cdot \vdash e_a : \tau_a, \text{ and } \tau = \tau_r \).
  From \( e \rightarrow_{cbv} e' \) and \( e = e_f e_a \), we have \( e_f e_a \rightarrow_{cbv} e' \).
  By cases on (the derivation of) \( e_f e_a \rightarrow_{cbv} e' \):
  - **E-APPLY** concludes the derivation of \( e_f e_a \rightarrow_{cbv} e' \):
    Therefore, \( e_f = \lambda x. e_b, e_a = v_a, \text{ and } e' = e_b[v_a/x] \).
    From \( e' = e_b[v_a/x] \) and \( e_a = v_a \), we have \( e' = e_b[e_a/x] \).
    From \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r, \cdot \vdash e_a : \tau_a, \text{ and } \tau = \tau_a \rightarrow \tau_r \).
    By inversion of \( \cdot \vdash \lambda x. e_b : \tau_a \rightarrow \tau_r \), we have \( \cdot, x : \tau_a \vdash e_b : \tau_r \).
    By **Substitution** applied to \( \cdot, x : \tau_a \vdash e_b : \tau_r \) and \( \cdot \vdash e_a : \tau_a \), we have \( \cdot \vdash e_b[e_a/x] : \tau_r \).
    From \( \cdot \vdash e_b[e_a/x] : \tau_r, e' = e_b[e_a/x], \text{ and } \tau = \tau_r \), we have \( \cdot \vdash e' : \tau_r \).

- **E-APPF** concludes the derivation of \( e_f e_a \rightarrow_{cbv} e' \):
  Therefore, \( e_f \rightarrow_{cbv} e'_f \) and \( e' = e'_f e_a \).
  By the induction hypothesis applied to \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r \) and \( e_f \rightarrow_{cbv} e'_f \), we have \( \cdot \vdash e'_f : \tau_a \rightarrow \tau_r \).
  From **T-APP**, \( \cdot \vdash e'_f : \tau_a \rightarrow \tau_r, \text{ and } \cdot \vdash e_a : \tau_a \),
  we can construct the derivation \( \cdot \vdash e'_f : \tau_a \rightarrow \tau_r, \cdot \vdash e_a : \tau_a \);
  therefore, we have \( \cdot \vdash e'_f e_a : \tau_r \).
  From \( \cdot \vdash e'_f e_a : \tau_r, e' = e'_f e_a, \text{ and } \tau = \tau_r \), we have \( \cdot \vdash e' : \tau_r \).

- **E-APPA** concludes the derivation of \( e_f e_a \rightarrow_{cbv} e' \):
  Therefore, \( e_a \rightarrow_{cbv} e'_a \) and \( e_f = v_f, \text{ and } e' = v_f e'_a \).
  From \( e' = v_f e'_a \) and \( e_f = v_f \), we have \( e' = e'_f e_a \).
  By the induction hypothesis applied to \( e_a \rightarrow_{cbv} e'_a \) and \( \cdot \vdash e_a : \tau_a \), we have \( \cdot \vdash e'_a : \tau_a \).
  From **T-APP**, \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r, \text{ and } \cdot \vdash e'_a : \tau_a \),
  we can construct the derivation \( \cdot \vdash e_f : \tau_a \rightarrow \tau_r, \cdot \vdash e'_a : \tau_a \);
  therefore, we have \( \cdot \vdash e_f e'_a : \tau_r \).
  From \( \cdot \vdash e_f e'_a : \tau_r, e' = e_f e'_a, \text{ and } \tau = \tau_r \), we have \( \cdot \vdash e' : \tau_r \).
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 $\text{T-Const}$, 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 $\Gamma, z : \tau_z \vdash x \sim \tau$.
  By cases on (the derivation of) $\Gamma, z : \tau_z \vdash x \sim \tau$.
    - C-Hit concludes the derivation of $\Gamma, z : \tau_z \vdash x \sim \tau$:
      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 $\Gamma, z : \tau_z \vdash x \sim \tau$:
      Therefore, $x \neq z$ and $\Gamma \vdash x \sim \tau$.
      From $\text{T-VAR}$, we can construct the derivation $\Gamma \vdash x \sim \tau$; 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 $\Gamma, z : \tau_z \vdash \lambda x. e_b : \tau$, we have $\Gamma, z : \tau_z, x : \tau_b \vdash e_b : \tau_b$ and $\tau = \tau_b \rightarrow \tau_r$.
  By $\text{Exchange}$ applied to $\Gamma, z : \tau_z, x : \tau_b \vdash e_b : \tau_b$ 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 $\Gamma, x : \tau_a \vdash e_b : \tau_z$.
  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 $\Gamma, x : \tau_a \vdash e_b[e_2/z] : \tau_r$; 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 $\Gamma \vdash e_1[e_2/z] : \tau$.

• $e_1 \equiv e_f e_a$:
  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 $e_1 = e_f e_a$, we have $\Gamma, z : \tau_z \vdash e_f e_a : \tau$.
  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 $\Gamma \vdash e_f[e_2/z] : \tau_a \rightarrow \tau_r, \Gamma \vdash e_a[e_2/z] : \tau_a$; therefore, we have $\Gamma \vdash e_f[e_2/z] e_a[e_2/z] : \tau_r$.
  By definition of substitution, we have $(e_f e_a)[e_2/z] = e_f[e_2/z] e_a[e_2/z]$.
  From $\Gamma \vdash e_f[e_2/z] e_a[e_2/z] : \tau_r, e_1 = e_f e_1, \tau = \tau_r$, and $(e_f e_a)[e_2/z] = e_f[e_2/z] e_a[e_2/z]$,
  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 $\Gamma, x : \tau_z \vdash e : \tau$.

**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 $e$.)