$$\Gamma \vdash \texttt{true} : \texttt{bool} \tag{1}$$

$$\Gamma \vdash \texttt{false} : \texttt{bool} \tag{2}$$

$$\Gamma \vdash x : \Gamma(x) \tag{3}$$

$$\frac{\Gamma \vdash e_1 : \texttt{bool} \qquad \Gamma \vdash e_2 : \tau \qquad \Gamma \vdash e_3 : \tau}{\Gamma \vdash \texttt{if}\ e_1\ \texttt{then}\ e_2\ \texttt{else}\ e_3 : \tau} \tag{4}$$

$$\frac{\Gamma \cup \{(x_1, \tau_1)\} \vdash e : \tau_2}{\Gamma \vdash \texttt{fn}(x) \Rightarrow e : \tau_1 \to \tau_2} \tag{5}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_1 \to \tau_2}{\Gamma \vdash e_1(e_2) : \tau_2} \tag{6}$$

$$\dfrac{\{\} \vdash \mathtt{true} : \mathtt{bool} \qquad \dfrac{\{(x, \tau_3)\} \vdash x : \tau_4}{\{\} \vdash \mathtt{fn}(x)\texttt{=>}x : \tau_3 \to t_4} \qquad \dfrac{\{(y, \tau_5)\} \vdash \mathtt{false} : \mathtt{bool}}{\{\} \vdash \mathtt{fn}(y)\texttt{=>}\mathtt{false} : \tau_5 \to \tau_6}}{\{\} \vdash \mathtt{if\ true\ then\ fn}(x)\texttt{=>}x\ \mathtt{else\ fn}(y)\texttt{=>}\mathtt{false} : \tau_1 \to \tau_2}$$

$$(\texttt{fn}(x)\texttt{=>}e)(v) \longrightarrow e[v/x] \tag{7}$$

$$\texttt{if } v\texttt{then } e_2 \texttt{ else } e_3 \longrightarrow \begin{cases} e_2 & \text{if } v = \texttt{true} \\ e_3 & \text{otherwise} \end{cases} \tag{8}$$

$$\frac{e_1 \longrightarrow e_1'}{e_1(e_2) \longrightarrow e_1'(e_2)} \tag{9}$$

$$\frac{e_2 \longrightarrow e_2'}{v_1(e_2) \longrightarrow v_1(e_2')} \tag{10}$$

$$\frac{e_1 \longrightarrow e_1'}{\texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 \longrightarrow \texttt{if } e_1' \texttt{ then } e_2 \texttt{ else } e_3} \tag{11}$$

Lemma (Substitution.)

If $\Gamma \cup \{(x, \tau')\} \vdash e : \tau$ *and* $\Gamma \vdash v : \tau'$*, then* $\Gamma \vdash e[v/x] : \tau$*.*

Theorem (Type Preservation.)

If $\Gamma \vdash e : \tau$ *and* $e \longrightarrow e'$*, then* $\Gamma \vdash e' : \tau$*.*

Lemma (Value Forms.)

If $\Gamma \vdash v : bool$*, then* v *is in the form* `true` *or* `false`*. If* $\Gamma \vdash v : \tau_1 \to \tau_2$*, then* v *is in the form* $fn(x) \Rightarrow e$*.*

Theorem (Progress.)

If e *is a closed expression and* $\Gamma \vdash e : \tau$*, then either* e *is a value or there exists an* e' *such that* $e \longrightarrow e'$*.*

Corollary (Soundness)

Well-typed programs cannot go wrong.

Substitution (Lemma 1) $\downarrow$ Type Preservation (Theorem 2) $\searrow$

Value Forms (Lemma 3) $\downarrow$ Progress (Theorem 4) $\swarrow$

Soundness (Corollary 5)

Lemma 1 (Substitution.) *If* $\Gamma \cup \{(x, \tau')\} \vdash e : \tau$ *and* $\Gamma \vdash v : \tau'$*, then* $\Gamma \vdash e[v/x] : \tau$*.*

Proof. *By induction on the derivation of* $\Gamma \cup \{(x, \tau')\} \vdash e : \tau$*.*
What was the last rule applied in order to derive $\Gamma \cup \{(x, \tau')\} \vdash e : \tau$*? Each possible rule is a different case, so we have a division into cases:*
Rule 1 or 2: *This would mean that* $e = c$*, where* c *is some boolean constant, and so* $\tau = \texttt{bool}$*. That is,* $\Gamma \cup \{x : \tau'\} \vdash c : \texttt{bool}$ *In other words,* x *doesn't even appear in* e*.*
Then $c[v/x] = c$ *and* $\Gamma \vdash c : \texttt{bool}$*.*

Rule 3: *Then* $e = y$ *for some variable* y. *(We choose* y *instead of* x *so the names don't clash.) Then we have two sub cases: either* y *and* x *really are the same variable, or* $y \neq x$.

> **Case a:** *Suppose* $x = y$. *Then* $\Gamma \cup \{x : \tau'\} \vdash x : \tau'$ *(that is,* $\tau = \tau'$*), by Rule 3. Moreover,* $x[v/x] = v$, *and* $\Gamma \vdash v : \tau'$.
>
> **Case b:** *Suppose* $x \neq y$. *Then the substitution doesn't change the expression at all.* $y[v/x] = y$ *and* $\Gamma \vdash y : \tau$.

Rule 4: *Suppose* $e = \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3$. *By Rule 4, it must be that* $\Gamma \cup \{x : \tau'\} \vdash e_1 : \texttt{bool}, e_2 : \tau, e_3 : \tau$. *Note that* $e[v/x] = \texttt{if } e_1[v/x] \texttt{ then } e_2 \texttt{ else } e_3[v/x]$. *By induction,* $\Gamma \vdash e_1[v/x] : \texttt{bool}, e_2[v/x] : \tau, e_3[v/x] : \tau$. *By Rule 4 again,*
$\Gamma \vdash \texttt{if } e_1[v/x] \texttt{ then } e_2 \texttt{ else } e_3[v/x] : \tau$.

Rule 5: *Suppose* $e = \texttt{fn}(y)\texttt{=>}e_1$. *Then* $\tau = \tau_1 \rightarrow t_2$ *for some* τ_1 *and* τ_2. *Note that*
$(\texttt{fn}(y)\texttt{=>}e_1)[v/x] = \texttt{fn}(y)\texttt{=>}e_1[v/x]$.
The last step of the derivation must have been

$$\frac{\Gamma \cup \{(x,\tau'),(y,\tau_1)\} \vdash e_1 : \tau_2}{\Gamma \cup \{(x,\tau')\} \vdash \texttt{fn}(y)\texttt{=>}e_1}$$

By induction, $\Gamma \cup \{(y,\tau_1)\} \vdash e_1[v/x] : \tau_2$. *By Rule 5,*
$\Gamma \vdash (\texttt{fn}(y)\texttt{=>}e_1)[v/x] : \tau_1 \rightarrow \tau_2$. *(Recall* $\tau = \tau_1 \rightarrow \tau_2$*.)*

Rule 6: *Suppose* $e = e_1(e_2)$*. Note that* $(e_1(e_2))[v/x] = e_1[v/x](e_2[v/x])$*.*
The last step of the derivation must have been

$$\frac{\Gamma \cup \{(x, \tau')\} \vdash e_1 : \tau_1 \to \tau \qquad \Gamma \cup \{(x, \tau')\} \vdash e_2 : \tau_1}{\Gamma \cup \{(x, \tau')\} \vdash e_2(e_2) : \tau}$$

for some τ_1*.*
By induction, $\Gamma \vdash e_1[v/x] : \tau_1 \to \tau, e_2[v/x] : \tau_1$*, and so by Rule 6,* $\Gamma \vdash e_1(e_2)[v/x] : \tau$*.*
Therefore, by examination of the cases, replacing x *with a value of the same type does not change the type of the expression.* $\square$

Theorem 2 (Type Preservation). *If* $\Gamma \vdash e : \tau$ *and* $e \longrightarrow e'$*, then* $\Gamma \vdash e' : \tau$*.*

Proof. *By induction on the derivation of* $\Gamma \vdash e : \tau$*.*
Rules 1 and 2: *Then* $e = c$ *for some constant* c*, so* $e \longrightarrow e'$ *is impossible.*
Rule 3: *Then* $e = x$ *for some variable* x*, so* $e \longrightarrow e'$ *is impossible.*
Rule 5: *Then* $e = \mathtt{fn}(x) \texttt{=>} e$*, so* $e \longrightarrow e'$ *is impossible.*

Rule 4: *Then* $e = \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3$. *This means the derivation is in the form*

$$\frac{\Gamma \vdash e_1 : \texttt{bool} \qquad \Gamma \vdash e_2 : \tau \qquad \Gamma \vdash e_3 : \tau}{\Gamma \vdash \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 : \tau}$$

e' *was derived either using Rule 8 or Rule 11.*

Rule 8: *Then* $e' = e_2$ *or* $e' = e_3$. *Either way, we've already shown that the type is* τ, *so* $\Gamma \vdash e' : \tau$.

Rule 11: *Then* $e_1 \longrightarrow e_1'$ *for some* e_1', *and also* $e' = \texttt{if } e_1' \texttt{ then } e_2 \texttt{ else } e_3$.

The important thing here is that $\Gamma \vdash e_1' : \texttt{bool}$.

Why is this true? Induction.

Now, applying rule 4 gives us $\Gamma \vdash \texttt{if } e_1' \texttt{ then } e_2 \texttt{ else } e_3 : \tau$.

Rule 6: *Then* $e = e_1(e_2)$. *That means the derivation is in the form*

$$\frac{\Gamma \vdash e_1 : \tau' \to \tau \qquad \Gamma \vdash e_2 : \tau'}{\Gamma \vdash e_1(e_2) : \tau}$$

for some τ'. *Now,* e' *was derived using one of Rule 7, 9, or 10.*

> **Rule 9:** *Then* $e_1 \longrightarrow e_1'$ *and* $e' = e_1'(e_2)$. *By induction,* $\Gamma \vdash e_1' : \tau' \to \tau$, *and by rule 6 we have* $\Gamma \vdash e_1'(e_2)$.
>
> **Rule 10** *is similar, just with* $e_2 \longrightarrow e_2'$.

Rule 7: *Then* e_1 *has the form* $\texttt{fn}(x) \texttt{=>} \hat{e}_1$*, and* e_2 *is a value, say* v_2*. Rule 7 ways that* $e' = \hat{e}_1[v_2/x]$*.*
Thus the derivation is

$$\frac{\dfrac{\Gamma \cup \{(x, \tau')\} \vdash \hat{e}_1 : \tau}{\Gamma \vdash e_1 : \tau' \to \tau} \qquad \Gamma \vdash v_2 : \tau'}{\Gamma \vdash (\texttt{fn}(x) \texttt{=>} \hat{e}_1)(v_2) : \tau}$$

Now we can apply Lemma 1. Since $\Gamma \cup \{(x, \tau')\} \vdash \hat{e}_1 : \tau$ *and* $\Gamma \vdash v_2 : \tau'$*, then we have* $\Gamma \vdash \hat{e}_1[v_2/x] : \tau$*.*

Therefore no matter what step is taken, the type is preserved. $\square$

Lemma 3 (Value Forms.) *If* $\Gamma \vdash v : bool$*, then* v *is in the form* $\mathtt{true}$ *or* $\mathtt{false}$*. If* $\Gamma \vdash v : \tau_1 \to \tau_2$*, then* v *is in the form* $fn(x) \Rightarrow e$*.*

Proof. *Immediate from rules 1, 2, and 5 and the definition of value.* □

Theorem 4 (Progress.) *If e is a closed expression and $\Gamma \vdash e : \tau$, then either e is a value or there exists an e' such that $e \longrightarrow e'$.*

Proof. *Once again, by induction on the derivation of $\Gamma \vdash e : \tau$. Once again, we divide this into cases based on the last rule applied in the derivation.*
Rules 1 and 2: *Then $e = c$, for some boolean constant c. Then e is a value.*
Rule 3: *Then $e = x$. This would mean x is a free variable, and e is not closed, contradicting our hypothesis. So, this case can't happen.*
Rule 5: *Then $e = \texttt{fn}(x)\texttt{=>}e_1$. Since e is closed, e is a value.*

Rule 4: *Then* $e = \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3$. *We need to show that, based on the information we have (specifically, it's closed and well-typed), that it can take a step.*
Since e is closed, so are e_1, e_2, *and* e_3.
The last step in the derivation was

$$\frac{\Gamma \vdash e_1 : \texttt{bool} \qquad \Gamma \vdash e_2 : \tau \qquad \Gamma \vdash e_3 : \tau}{\Gamma \vdash \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 : \tau}$$

e_1 *is either value or it is not a value.*

Case 1: *Suppose* e_1 *is a value. Then, since* $\Gamma \vdash e_1 : \texttt{bool}$, *then by lemma 3,* e_1 *is either* true or false. *Hence by rule 8, either* $e \longrightarrow e_1$ *or* $e \longrightarrow e_2$.

Case 2: *Suppose* e_1 *is not a value. Then* $e_1 \longrightarrow e_1'$ *for some* e_1' *by structural induction. Then we apply rule 11:* $e \longrightarrow \texttt{if } e_1' \texttt{ then } e_2 \texttt{ else } e_3$.

Rule 6: $e = e_1(e_2)$. *Since* e *is closed,* e_1 *and* e_2 *also are closed.*
The last step in the derivation is

$$\frac{\Gamma \vdash e_1 : \tau' \to \tau \qquad \Gamma \vdash e_2 : \tau'}{\Gamma \vdash e_1(e_2) : \tau}$$

for some τ'. *By induction,* e_1 *and* e_2 *are each either values or they reduce to another expression.*
We want to show that under the given circumstances, e *can take a step. There are three cases:*

- e_1 *and* e_2 *are both values.*
- e_1 *is not a value (*e_2*, maybe, maybe not).*
- e_1 *is a value but* e_2 *is not.*

Case 1: *Suppose* e_1 *and* e_2 *are both values. By lemma 3,* e_1 *has the form* $\texttt{fn}(x){=}{>}\hat{e}_1$*. Then* $e_1(e_2) = (\texttt{fn}(x){=}{>}\hat{e}_1)(e_2) \longrightarrow \hat{e}_1[e_2/x]$ *by rule 7.*

Case 2: *Suppose* e_1 *is not a value. By induction, there exists* e_1' *such that* $e_1 \longrightarrow e_1'$*, so by rule 9,* $e_1(e_2) \longrightarrow e_1'(e_2)$*.*

Case 3: *Suppose* e_1 *is a value but* e_2 *is not. By induction,* $e_2 \longrightarrow e_2'$ *for some* e_2'*. By rule 10,* $e_1(e_2) \longrightarrow e_1(e_2')$*.*

□

Corollary 5 (Soundness.) *Well-typed programs cannot go wrong.*

Proof. *Combine Theorems 2 and 4. Specifically, suppose* $\Gamma \vdash e : \tau$.
Then, by Theorem 4, either e *is a value or* $e \longrightarrow e'$ *for some* e'. *In the latter case, Theorem 2 says that* $\Gamma \vdash e' : \tau$. *Then apply Corollary 5 inductively.* $\square$