

$$\Gamma \vdash \mathbf{true} : \mathbf{bool} \quad (1)$$

$$\Gamma \vdash \mathbf{false} : \mathbf{bool} \quad (2)$$

$$\Gamma \vdash x : \Gamma(x) \quad (3)$$

$$\frac{\Gamma \vdash e_1 : \mathbf{bool} \quad \Gamma \vdash e_2 : \tau \quad \Gamma \vdash e_3 : \tau}{\Gamma \vdash \mathbf{if } e_1 \mathbf{ then } e_2 \mathbf{ else } e_3 : \tau} \quad (4)$$

$$\frac{\Gamma \cup \{(x_1, \tau_1)\} \vdash e : \tau_2}{\Gamma \vdash \mathbf{fn}(x) \Rightarrow e : \tau_1 \rightarrow \tau_2} \quad (5)$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_1 \rightarrow \tau_2}{\Gamma \vdash e_1(e_2) : \tau_2} \quad (6)$$

We say that an expression e is *well-typed* in a type system if there exists an environment Γ and a type τ such that the judgment $\Gamma \vdash e : \tau$ can be proven by the type rules.

A type system is *sound* if well-typed programs cannot cause type errors.

What we want to do is prove BoolEm's type system to be sound.

An expression is *closed* if it has no free variables.

A *program* is a closed expression.

A *value* is a closed abstraction or a boolean constant. We'll use the value v to range over values.

$$(\text{fn}(x) \Rightarrow e)(v) \longrightarrow e[v/x] \quad (7)$$

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

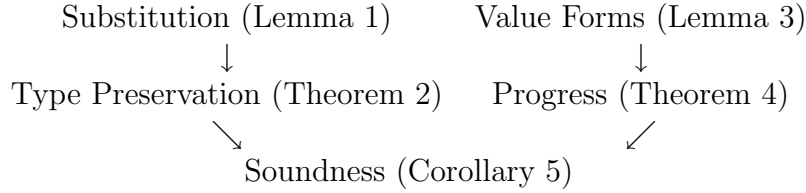
$$\frac{e_1 \longrightarrow e'_1}{e_1(e_2) \longrightarrow e'_1(e_2)} \quad (9)$$

$$\frac{e_2 \longrightarrow e'_2}{v_1(e_2) \longrightarrow v_1(e'_2)} \quad (10)$$

$$\frac{e_1 \longrightarrow e'_1}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 \longrightarrow \text{if } e'_1 \text{ then } e_2 \text{ else } e_3} \quad (11)$$

An expression e is *stuck* if it is not a value and there does not exist an e' such that $e \longrightarrow e'$.

An expression *goes wrong* if it evaluates to a stuck expression.



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

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

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

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 $e \longrightarrow e'$.*

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