Big Step Semantics

\[ e \Rightarrow e' \] "e steps to e'" (one step)

\[ e \Rightarrow^* e' \] "e steps to e'" (many steps)

\[ e \Rightarrow^* v \text{ and } v \text{ val} \] "e evaluates to v"

\[ e \downarrow v \] "e evaluates to v"

\[ e := \lambda x.e \mid e \mid (e,e) \mid \text{fst } e \mid \text{snd } e \mid \text{linr } e \mid \text{linl } e \mid \text{inr } e \mid \text{inl } e \mid \text{fix } x = e \]

\[ \tau := \text{wit } \tau \rightarrow \tau \mid \tau < \tau \mid \tau + \tau \]

\[ v := () \mid \lambda x.e \mid (v,v) \mid \text{linl } v \mid \text{linr } v \]

Combines 3 step rules for application

\[ \frac{}{v \downarrow v} \quad (E\text{-}1) \quad \frac{e \downarrow x.e \quad e' \downarrow v'}{e_{x.e} \downarrow v'} \quad (E\text{-}2) \quad \frac{v \downarrow v}{e \downarrow v} \]

\[ \frac{e \downarrow v_1 \quad e' \downarrow v_2}{(e,e) \downarrow (v_1,v_2)} \quad (E\text{-}3) \quad \frac{v \downarrow (v_1,v_2)}{\text{fst } e \downarrow v_1} \quad (E\text{-}4) \quad \frac{v \downarrow (v_1,v_2)}{\text{snd } e \downarrow v_2} \quad (E\text{-}5) \]

\[ \frac{e \downarrow v}{\text{linl } e \downarrow v} \quad (E\text{-}6) \quad \frac{e \downarrow v}{\text{linr } e \downarrow v} \quad (E\text{-}7) \]

\[ \frac{e \downarrow v \quad (v,k)e \downarrow v'}{(v_{(x)}e)\downarrow v} \quad (E\text{-}8) \quad \frac{e \downarrow \text{inr } v}{(v_{(x)}e)\downarrow v} \quad (E\text{-}9) \]

\[ \text{Way less rules!} \]

\[ \frac{\text{case } e \text{ of } e_1 \mid e_2}{(e_{\text{case } e_1, e_2}) \downarrow v} \quad (E\text{-}10) \]

\[ \frac{\text{case } e_1 \text{ of } \text{ex } e_2, y, z}{(e_{\text{case } e_1, y, z}) \downarrow v} \quad (E\text{-}11) \]
Preservation: If $\Gamma \vdash e : \tau$ and $e_1, e_2 \Downarrow \tau$, then $\Gamma \vdash e_1 \Downarrow \tau \land e_2 \Downarrow \tau$.

**Proof:**

1. **(E1)** $\checkmark$
2. **(E2)** Then $e = e_1, e_2$ and $e_1 \Downarrow \tau \land e_2 \Downarrow \tau$ and $[\tau/\lambda x]e \Downarrow \tau$.
   By inversion, $\tau = \tau_1 \land \tau_2$ and $\Gamma \vdash e_1 : \tau_1 \rightarrow \tau_2$ and $\Gamma \vdash e_2 : \tau_1$.
   By induction, $\Gamma \vdash \lambda x.e : \tau_1 \rightarrow \tau_2$ and $\Gamma \vdash e : \tau_2$.
   By inversion, $\Gamma, x : \tau \Downarrow e : \tau_2$.
   By substitution, $\Gamma \vdash \lambda [\tau/\lambda x]e : \tau_2$.
   By induction, $\Gamma \vdash e : \tau_2$.

3. **(E3)** Then $e = (e_1, e_2)$ and $e_1 \Downarrow \tau_1$ and $e_2 \Downarrow \tau_2$ and $\nu = (\nu, \nu')$.
   By inversion, $\tau = \tau_1 \times \tau_2$ and $\Gamma \vdash e_1 : \tau_1$ and $\Gamma \vdash e_2 : \tau_2$.
   By induction, $\Gamma \vdash \nu : \tau_1$ and $\Gamma \vdash \nu : \tau_2$.
   By typing rules, $\Gamma \vdash \nu : \tau_1 \times \tau_2$.

4. **(E4)** Then $e = \text{fst} e_0$ and $e_0 \Downarrow (\nu, \nu')$.
   By inversion, $\tau = \tau_1 \times \tau_2$ and $\Gamma \vdash e_0 : \tau_1$.
   By induction, $\Gamma \vdash \nu : \tau_1$.

5. **(E5)** Then $e = \text{in1} e_0$ and $e_0 \Downarrow \nu$ and $\nu = \text{in1} \nu'$.
   By inversion, $\tau = \tau_1 + \tau_2$ and $\Gamma \vdash e_0 : \tau_1$.
   By induction, $\Gamma \vdash \nu : \tau_1$. By typing rules, $\Gamma \vdash \nu : \tau_1 + \tau_2$.

6. **(E6)** Then $e = \text{case} e_1$ of $\text{ex} e_2, x. e_3$ and $e_1 \Downarrow \nu$ and $\nu = \text{in1} \nu'$ and $\nu' = \text{in1} \nu$.
   By inversion, $\tau = \tau_1 + \tau_2$ and $\Gamma \vdash e_1 : \tau_1$.
   By induction, $\Gamma \vdash \nu : \tau_1$. By typing rules, $\Gamma \vdash \nu : \tau_1 + \tau_2$.

7. **(E8)** Then $e = \text{case} e_1$ of $\text{ex} e_2, x. e_3$ and $e_1 \Downarrow \nu$ and $\nu = \text{in1} \nu'$ and $\nu' = \text{in1} \nu$.
   By inversion, $\tau = \tau_1 + \tau_2$ and $\Gamma \vdash e_1 : \tau_1$.
   By induction, $\Gamma \vdash \nu : \tau_1$. By typing rules, $\Gamma \vdash \nu : \tau_1 + \tau_2$.

8. **(E9)** Then $e = \text{fix} x. e_0$ and $e \Downarrow e_0$.
   By in, $\Gamma, x : \tau \Downarrow e : \tau$.

9. **(E10)** Then $e = \text{fix} x. e_0$ and $\text{let} x = e_0(x) e_0 \Downarrow \nu$.
   By in, $\Gamma, x : \tau \Downarrow e : \tau$.

By substitution and induction, $\Gamma \vdash e : \tau$. 

Progress: ...
One option: If \( f \models e \), then \( \exists v \text{ s.t. } e \ll v \)

True in STLC w/o \( \text{fix} \) but not in most real languages.

\[ \exists v. \text{fix } x = x \cup v. \]

Big step can't talk about non-terminating expressions
No real way to talk about progress (\( \Rightarrow \) type safety).

But: - Don't need to worry about evaluation order
    - Don't need all the search rules

Thm. \( e \rightarrow^* v \iff e \ll v \)
    \( \iff \) Fairly straightforward with a couple annoying lemmas
    \( \Rightarrow \) Substitutes to show

Lemma. If \( e \rightarrow e' \) and \( e' \ll v \) then \( e \ll v \).