We defined the semantics of MinHs using two different approaches:

- control flow implicit: M-machine
- control flow explicit: C-machine, E-machine

Are they equivalent?

Proof is not obvious, as evaluation methods are very different.

How can we make the connection?

Observation:

1. A single M-machine step usually corresponds to many C-machine steps.
2. The stack in C-machine corresponds to context of subexpression which is actually evaluated in M-machine.
3. Stack and current expression provide enough information to reconstruct unevaluated portion of original expression.

Reconstruction:

\[
\begin{align*}
\circ \circ_i & = e \\
(\text{plus}(\mathbb{E}, e_2) \circ s) @ e_1 & = s @ \text{plus}(e_1, e_2) \\
(\text{plus}(e_1, \mathbb{E}) \circ s) @ e_2 & = s @ \text{plus}(e_1, e_2) \\
(\text{if}(\mathbb{E}, e_1, e_2) \circ s) @ e_c & = s @ \text{if}(e, e_1, e_2) \\
\end{align*}
\]

Show that

1. If \( s @ e \rightarrow C^* \circ s < v \), then \( e @ s \rightarrow M^* v \)

Note: we show that the above is true for any stack \( s \) (as opposed to the empty stack), since the more general statement is easier to prove as the i.h. is stronger.

Here and in the remainder of the proof we write \( s @ e \rightarrow \) to mean \( \rightarrow \) or \( \leftarrow \).

2. If \( e \rightarrow M v \), then \( e @ e \rightarrow C v \) \( s @ e \leftarrow v \)

Both directions can be proven using induction over the number of evaluation steps.
- **Base case**: in the C-machine, the minimum number of steps necessary to evaluate an expression is one. Show that, if \( s \triangleright e \xrightarrow{C} o \triangleleft e \) then \( s \notin e \xrightarrow{C} v \).

  The expression has to be a value already — no evaluation step necessary in the M-machine.

- **Induction step**: \( n + 1 \) evaluation steps necessary.

  \[
  s \triangleright e \xrightarrow{C} e' \xrightarrow{C} o \triangleleft v
  \]

  **Induction Hypothesis**:

  \[
  e' \xrightarrow{C} e' \xrightarrow{C} v
  \]

  We need to prove the following lemma:

  **Lemma 1**: If \( s \triangleright e \xrightarrow{C} s' \triangleright e' \), then either \( s \notin e = s' \triangleleft e' \) or \( s \notin e \xrightarrow{M} s' \triangleleft e' \)

---

**Slide 6**

- **Base case**: Show that, if \( e \xrightarrow{M} v \) then \( o \triangleright e \xrightarrow{M} o \triangleleft v \).

- **Induction step**: \( n + 1 \) evaluation steps necessary.

  \[
  e \xrightarrow{M} e' \xrightarrow{M} v
  \]

  **Induction Hypothesis**:

  \[
  o \triangleright e' \xrightarrow{C} o \triangleleft v
  \]

  Again, we first need to prove a lemma:

  **Lemma 2**: If \( e \xrightarrow{M} e' \) and \( s \triangleright e' \xrightarrow{C} o \triangleleft v \) then \( s \triangleright e \xrightarrow{C} o \triangleleft v \).

---

Both lemmata can be shown by rule induction over the definition of \( \xrightarrow{C} \) and \( \xrightarrow{M} \) respectively (see textbook).