By popular demand, here are proofs for theorems only sketched in the lecture-5 slides. (The equivalence proof for small-step and large-step expression semantics is complete enough in the slides I think. I wrote these very quickly, so corrections are welcome.

Theorem: \( H ; e \star 2 \Downarrow c \) if and only if \( H ; e + e \Downarrow c \).

Proof: (Does not use induction)

- First assume \( H ; e \star 2 \Downarrow c \) and show \( H ; e + e \Downarrow c \). Any derivation of \( H ; e \star 2 \Downarrow c \) must end with the \textsc{mult} rule, which means there must exist derivations of \( H ; e \Downarrow c' \) and \( H ; 2 \Downarrow 2 \), and \( c \) must be \( 2c' \). That is, there must be a derivation that looks like this:

\[
\frac{H ; e \Downarrow c'}{H ; e \star 2 \Downarrow 2c'}
\]

So given that there exists a derivation of \( H ; e \Downarrow c' \), we can use \textsc{add} to derive:

\[
\frac{H ; e \Downarrow c' \quad H ; e \Downarrow c'}{H ; e + e \Downarrow c' + c'}
\]

Math provides \( c' + c' = 2c' \), so the conclusion of this derivation is what we need.

- Now assume \( H ; e + e \Downarrow c \) and show \( H ; e \star 2 \Downarrow c \). Any derivation of \( H ; e + e \Downarrow c \) must end with the \textsc{add} rule, which means there exists a derivation that looks like this (where \( c = c_1 + c_2 \)):

\[
\frac{H ; e \Downarrow c_1 \quad H ; e \Downarrow c_2}{H ; e + e \Downarrow c_1 + c_2}
\]

In fact, we earlier proved determinacy (there is at most one \( c \) such that \( H ; e \Downarrow c \)), so the derivation must have this form (where \( c = c_1 + c_1 \)):

\[
\frac{H ; e \Downarrow c_1 \quad H ; e \Downarrow c_1}{H ; e + e \Downarrow c_1 + c_1}
\]

So given that there exists a derivation of \( H ; e \Downarrow c_1 \), we can use \textsc{mult} to derive:

\[
\frac{H ; e \Downarrow c_1 \quad H ; 2 \Downarrow 2}{H ; e \star 2 \Downarrow 2c_1}
\]

Math provides \( c_1 + c_1 = 2c_1 \), so the conclusion of this derivation is what we need.
\[ C ::= \mathbb{[} \mid C + e \mid e + C \mid C \ast e \mid e \ast C \]

Formal definition of “filling the hole”:

\[
\begin{align*}
(\mathbb{[}]|e) &= e \\
(C + e_1)|e) &= C[e] + e_1 \\
(e_1 + C)|e) &= e_1 + C[e] \\
(C \ast e_1)|e) &= C[e] \ast e_1 \\
(e_1 \ast C)|e) &= e_1 \ast C[e]
\end{align*}
\]

Theorem: \( H : C'[e + 2] \Downarrow c \) if and only if \( H : C[e + e] \Downarrow c \).

Proof: By induction on (the height of) the structure of \( C \):

- If the height is 1, then \( C = \mathbb{[} \), so \( C[e + 2] = e + 2 \) and \( C[e + e] = e + e \). So the previous theorem is exactly what we need.

- If the height is greater than 1, then \( C \) has one of four forms:
  - If \( C \) is \( C' + e' \) for some \( C' \) and \( e' \), then \( C[e + 2] \) is \( C'[e + 2] + e' \) and \( C[e + e] \) is \( C'[e + e] + e' \). Since \( C' \) is shorter than \( C \), induction ensures that for any constant \( c' \), \( H : C'[e + 2] \Downarrow c' \) if and only if \( H : C'[e + e] \Downarrow c' \).
    Assume \( H : C'[e + 2] + e' \Downarrow c \) and show \( H : C'[e + e] + e' \Downarrow c \): Any derivation of \( H : C'[e + 2] + e' \Downarrow c \) must end with \( \text{ADD} \), i.e., it looks like this (where \( c = c' + c'' \)):
      \[
      \begin{array}{c}
      \vdots \\
      H : C'[e + 2] \Downarrow c' \\
      H : e' \Downarrow c'' \\
      \hline
      H : C'[e + 2] + e' \Downarrow c
      \end{array}
      \]
    As argued above, the existence of a derivation of \( H : C'[e + 2] \Downarrow c' \) ensures the existence of a derivation of \( H : C'[e + e] \Downarrow c' \). So using \( \text{ADD} \) and the existence of a derivation of \( H : e' \Downarrow c'' \), we can derive:
    \[
    \begin{array}{c}
    \vdots \\
    H : C'[e + e] \Downarrow c' \\
    H : e' \Downarrow c'' \\
    \hline
    H : C'[e + 2] + e' \Downarrow c
    \end{array}
    \]
  - The other 3 cases are similar. (Try them out.)
Theorem: Informally, the statement-sequence operator is associative. Formally:

(a) For all \( n \), if \( H ; s_1; (s_2; s_3) \rightarrow^n H' ; \text{skip} \) then there exist \( H' \) and \( n' \) such that \( H ; (s_1; s_2); s_3 \rightarrow^n H'' ; \text{skip} \) and \( H''(\text{ans}) = H'(\text{ans}) \).

(b) If for all \( n \) there exist \( H' \) and \( s' \) such that \( H ; s_1; (s_2; s_3) \rightarrow^n H' ; s' \), then for all \( n \) there exist \( H'' \) and \( s'' \) such that \( H ; (s_1; s_2); s_3 \rightarrow^n H'' ; s'' \).

Lemma: For all \( n \), if \( H ; s_1; (s_2; s_3) \rightarrow^n H' ; s' \), then either (1) \( s' \) has the form \( s'_1; (s_2; s_3) \) and \( H ; (s_1; s_2); s_3 \rightarrow^n H' ; (s'_1; s_2); s_3 \) or (2) \( H ; (s_1; s_2); s_3 \rightarrow^n H'' ; s'' \).

Proof of the lemma: By induction on \( n \). For the base case \( n = 0 \), so (1) holds with \( s'_1 = s_1 \). For the inductive case \( n > 0 \), so \( H ; s_1; (s_2; s_3) \rightarrow^n H' ; s' \), which means \( H ; s_1; (s_2; s_3) \rightarrow^{n-1} H'' ; s'' \) and \( H'' ; s'' \rightarrow H' ; s' \) for some \( H'' \) and \( s'' \). So by induction either (1) \( s'' \) has the form \( s''_1; (s_2; s_3) \) and \( H ; (s_1; s_2); s_3 \rightarrow^{n-1} H'' ; (s''_1; s_2); s_3 \) or (2) \( H ; (s_1; s_2); s_3 \rightarrow^{n-1} H'' ; s'' \).

If (1), then the derivation of \( H'' ; s'' \rightarrow H' ; s' \) ends with either Seq1 or Seq2. If Seq1, then \( H'' \) is \( H' \), \( s''_1 \) is skip and \( s' \) is \( s_2; s_3 \). Furthermore, we can derive:

\[
\frac{H'' ; \text{skip}; s_2 \rightarrow H'' ; s_2}{H'' ; (\text{skip}; s_2); s_3 \rightarrow H'' ; s_2; s_3}
\]

So (2) holds. If Seq2, then the derivation of \( H'' ; s'' \rightarrow H' ; s' \) must have the form:

\[
\frac{H'' ; s'_1 \rightarrow H' ; s'_1}{H'' ; s_1; (s_2; s_3) \rightarrow H' ; s'_1; (s_2; s_3)}
\]

So there must be a derivation of \( H'' ; s'_1 \rightarrow H' ; s'_1 \). So we can derive:

\[
\frac{H'' ; s''_1 \rightarrow H' ; s'_1}{H'' ; (s''_1; s_2) \rightarrow H' ; s'_1; s_2}
\]

So (1) holds.

If (2), then \( H'' ; s'' \rightarrow H' ; s' \) ensures \( H ; (s_1; s_2); s_3 \rightarrow H' ; s' \), so (2) holds.