Figure 19.1: Dynamic environment semantics rules for evaluating expressions, for a functional language with naming and arithmetic.

$$E \vdash \overline{n} \Downarrow \overline{n} \qquad (R_{int})$$

$$E \vdash x \Downarrow E(x) \qquad (R_{var})$$

$$E \vdash \texttt{fun}\ x\ \texttt{->}\ P \Downarrow \texttt{fun}\ x\ \texttt{->}\ P \qquad (R_{fun})$$

$$\begin{array}{l} E \vdash P\ \texttt{+}\ Q \Downarrow \\ \qquad \left| \begin{array}{l} E \vdash P \Downarrow \overline{m} \\ E \vdash Q \Downarrow \overline{n} \end{array} \right. \\ \qquad \Downarrow \overline{m+n} \end{array} \qquad (R_{+})$$

(and similarly for other binary operators)

$$\begin{array}{l} E \vdash \texttt{let}\ x\ \texttt{=}\ D\ \texttt{in}\ B \Downarrow \\ \qquad \left| \begin{array}{l} E \vdash D \Downarrow v_D \\ E\{x \mapsto v_D\} \vdash B \Downarrow v_B \end{array} \right. \\ \qquad \Downarrow v_B \end{array} \qquad (R_{let})$$

$$\begin{array}{l} E \vdash P\ \ Q \Downarrow \\ \qquad \left| \begin{array}{l} E \vdash P \Downarrow \texttt{fun}\ x\ \texttt{->}\ B \\ E \vdash Q \Downarrow v_Q \\ E\{x \mapsto v_Q\} \vdash B \Downarrow v_B \end{array} \right. \\ \qquad \Downarrow v_B \end{array} \qquad (R_{app})$$

Figure 19.2: Lexical environment semantics rules for evaluating expressions, for a functional language with naming and arithmetic.

$$E \vdash \overline{n} \Downarrow \overline{n} \qquad (R_{int})$$

$$E \vdash x \Downarrow E(x) \qquad (R_{var})$$

$$E \vdash \texttt{fun}\ x\ \texttt{->}\ P \Downarrow [E \vdash \texttt{fun}\ x\ \texttt{->}\ P] \qquad (R_{fun})$$

$$\begin{array}{l} E \vdash P\ \texttt{+}\ Q \Downarrow \\ \qquad \left| \begin{array}{l} E \vdash P \Downarrow \overline{m} \\ E \vdash Q \Downarrow \overline{n} \end{array} \right. \\ \qquad \Downarrow \overline{m+n} \end{array} \qquad (R_{+})$$

(and similarly for other binary operators)

$$\begin{array}{l} E \vdash \texttt{let}\ x\ \texttt{=}\ D\ \texttt{in}\ B \Downarrow \\ \qquad \left| \begin{array}{l} E \vdash D \Downarrow v_D \\ E\{x \mapsto v_D\} \vdash B \Downarrow v_B \end{array} \right. \\ \qquad \Downarrow v_B \end{array} \qquad (R_{let})$$

$$\begin{array}{l} E \vdash \texttt{let rec}\ x\ \texttt{=}\ D\ \texttt{in}\ B \Downarrow \\ \qquad \left| \begin{array}{l} E\{x \mapsto \texttt{let rec}\ x\ \texttt{=}\ D\ \texttt{in}\ x\} \vdash D \Downarrow v_D \\ E\{x \mapsto v_D\} \vdash B \Downarrow v_B \end{array} \right. \\ \qquad \Downarrow v_B \end{array} \qquad (R_{letrec})$$

$$\begin{array}{l} E_d \vdash P\ \ Q \Downarrow \\ \qquad \left| \begin{array}{l} E_d \vdash P \Downarrow [E_l \vdash \texttt{fun}\ x\ \texttt{->}\ B] \\ E_d \vdash Q \Downarrow v_Q \\ E_l\{x \mapsto v_Q\} \vdash B \Downarrow v_B \end{array} \right. \\ \qquad \Downarrow v_B \end{array} \qquad (R_{app})$$

Figure 19.4: Lexical environment semantics rules for evaluating expressions, for a functional language with naming and arithmetic.

$$E, S \vdash \overline{n} \Downarrow \overline{n}, S \qquad (R_{int})$$

$$E, S \vdash x \Downarrow E(x), S \qquad (R_{var})$$

$$E, S \vdash \texttt{fun}\ x\ \texttt{->}\ P \Downarrow [E \vdash \texttt{fun}\ x\ \texttt{->}\ P], S \qquad (R_{fun})$$

$$\begin{array}{l} E, S \vdash P\ \texttt{+}\ Q \Downarrow \\ \qquad \left| \begin{array}{l} E, S \vdash P \Downarrow \overline{m}, S' \\ E, S' \vdash Q \Downarrow \overline{n}, S'' \end{array} \right. \\ \qquad \Downarrow \overline{m+n}, S'' \end{array} \qquad (R_{+})$$

(and similarly for other binary operators)

$$\begin{array}{l} E, S \vdash \texttt{let}\ x\ \texttt{=}\ D\ \texttt{in}\ B \Downarrow \\ \qquad \left| \begin{array}{l} E, S \vdash D \Downarrow v_D, S' \\ E\{x \mapsto v_D\}, S' \vdash B \Downarrow v_B, S'' \end{array} \right. \\ \qquad \Downarrow v_B, S'' \end{array} \qquad (R_{let})$$

$$\begin{array}{l} E, S \vdash \texttt{let rec}\ x\ \texttt{=}\ D\ \texttt{in}\ B \Downarrow \\ \qquad \left| \begin{array}{l} E\{x \mapsto \texttt{let rec}\ x\ \texttt{=}\ D\ \texttt{in}\ x\}, S \vdash D \Downarrow v_D, S' \\ E\{x \mapsto v_D\}, S' \vdash B \Downarrow v_B, S'' \end{array} \right. \\ \qquad \Downarrow v_B, S'' \end{array} \qquad (R_{letrec})$$

$$\begin{array}{l} E_d, S \vdash P\ \ Q \Downarrow \\ \qquad \left| \begin{array}{l} E_d, S \vdash P \Downarrow [E_l \vdash \texttt{fun}\ x\ \texttt{->}\ B], S' \\ E_d, S' \vdash Q \Downarrow v_Q, S'' \\ E_l\{x \mapsto v_Q\}, S'' \vdash B \Downarrow v_B, S''' \end{array} \right. \\ \qquad \Downarrow v_B, S''' \end{array} \qquad (R_{app})$$

$$\begin{array}{l} E, S \vdash \texttt{ref}\ P \Downarrow \\ \qquad \left| \begin{array}{l} E, S \vdash P \Downarrow v_P, S' \end{array} \right. \\ \qquad \Downarrow l, S'\{l \mapsto v_P\} \quad \text{(where } l \text{ is a new location)} \end{array} \qquad (R_{ref})$$

$$\begin{array}{l} E, S \vdash \texttt{!}\ P \Downarrow \\ \qquad \left| \begin{array}{l} E, S \vdash P \Downarrow l, S' \end{array} \right. \\ \qquad \Downarrow S'(l), S' \end{array} \qquad (R_{deref})$$

$$\begin{array}{l} E, S \vdash P\ \texttt{:=}\ Q \Downarrow \\ \qquad \left| \begin{array}{l} E, S \vdash P \Downarrow l, S' \\ E, S' \vdash Q \Downarrow v_Q, S'' \end{array} \right. \\ \qquad \Downarrow (), S''\{l \mapsto v_Q\} \end{array} \qquad (R_{assign})$$

$$\begin{array}{l} E, S \vdash P\ \texttt{;}\ Q \Downarrow \\ \qquad \left| \begin{array}{l} E, S \vdash P \Downarrow (), S' \\ E, S' \vdash Q \Downarrow v_Q, S'' \end{array} \right. \\ \qquad \Downarrow v_Q, S'' \end{array} \qquad (R_{seq})$$