$$
\begin{array}{lcl}
bop & ::= & \textbf{+} \mid \textbf{-} \mid \textbf{*} \mid \textbf{/} \mid \textbf{¡} \mid \cdots \\
op & ::= & \textbf{transpose} \mid \textbf{rearrange}\ (d, \cdots, d) \mid \textbf{replicate} \\
soac & ::= & \textbf{map} \mid \textbf{reduce} \mid \textbf{scan} \mid \textbf{redomap} \mid \textbf{scanomap} \\
e & ::= & x \mid d \mid b \mid (e, \cdots, e) \mid e[e] \mid e\ bop\ e \mid op\ e\ \cdots\ e \\
 & \mid & \textbf{loop}\ x_1\ \cdots\ x_n \textbf{ = } e\ \textbf{for}\ y < e\ \textbf{do}\ e \\
 & \mid & \textbf{loop}\ x_1\ \cdots\ x_n \textbf{ = } e\ \textbf{for}\ y = e\ \textbf{to}\ 0\ \textbf{do}\ e \\
 & \mid & \textbf{let}\ x_1\ \cdots\ x_n \textbf{ = } e\ \textbf{in}\ e \mid \textbf{if}\ e\ \textbf{then}\ e\ \textbf{else}\ e \\
 & \mid & soac\ f\ e\ \cdots\ e \\
f & ::= & \lambda x_1\ \cdots\ x_n \rightarrow e \mid soac\ f\ e\ \cdots\ e \mid e\ bop \mid bop\ e
\end{array}
$$

Reverse-mode Rules

We define *tape maps* ($\parallel \Omega$) and *adjoint contexts* ($\Lambda \vdash$) as

$$
\begin{array}{l}
\Omega ::= \varepsilon \mid \Omega, (x \mapsto x_s) \\
\Lambda ::= \varepsilon \mid \Lambda, (x \mapsto \hat{x})
\end{array}
$$

Forward pass ($\Rightarrow_F$)

$$
\frac{e = \textbf{loop}\ \overline{x} \textbf{ = } e_0\ \textbf{for}\ y < e_n\ \textbf{do}\ e_{body} \qquad x_{s_0}\ \text{fresh} \qquad x_{s_0} = \textbf{replicate}\ e_n\ \textbf{0}}{e \Rightarrow_F \textbf{loop}\ (\overline{x}, x_s) \textbf{ = } (e_0, x_{s_0})\ \textbf{for}\ y < e_n\ \textbf{do}\ (e_{body}, x_s[y] = \overline{x}) \parallel (x \mapsto x_{s_0})}\ \textsc{FwdLoop}
$$

$$
\frac{\textbf{if}\ e_p\ \textbf{then}\ e_t\ \textbf{else}\ e_f e_t \Rightarrow_F e'_t \parallel \Omega_t e_f \Rightarrow_F e'_f \parallel \Omega_f}{\textbf{if}\ e_p\ \textbf{then}\ e_t\ \textbf{else}\ e_f \Rightarrow_F \textbf{if}\ e_p\ \textbf{then}\ e'_t\ \textbf{else}\ e'_f \parallel \Omega_t \cup \Omega_f}\ \textsc{FwdIFE}
$$

Reverse pass ($\Lleftarrow$)

$$
\frac{\begin{array}{c}
e_{body} = \textbf{let}\ \overline{rs} \textbf{ = } e'_{body}\ \textbf{in}\ \overline{rs} \qquad e_{loop} = \textbf{let}\ \overline{lres} \textbf{ = loop}\ \overline{x} \textbf{ = } e_0\ \textbf{for}\ y < e_n\ \textbf{do}\ e_{body}\ \textbf{in}\ \overline{lres} \\
e_{loop} \Rightarrow_F e'_{loop} \parallel \Omega \qquad \overline{fv} = FV(e_{body}) \setminus \overline{x} \\
\hat{\overline{x}},\ \overline{\hat{fv}},\ \overline{\hat{rs}}\ fresh \qquad reset = \textbf{map}\ (\lambda \_ .\ \textbf{0})\ \hat{\overline{x}} \qquad \Lambda'_1 = \Lambda_1,\ \overline{x} \mapsto \hat{\overline{x}},\ \overline{fv} \mapsto \overline{\hat{fv}},\ \overline{rs} \mapsto \overline{\hat{rs}} \\
\hat{e}_{body} = \textbf{let}\ \overline{\hat{rs}'} \textbf{ = } \hat{e}'_{body}\ \textbf{in}\ \overline{\hat{rs}'} \qquad (\Lambda'_1 \vdash e_{body}) \Lleftarrow (\Lambda_2 \vdash \hat{e}_{body}) \\
\hat{e}'_{body} = \textbf{let}\ \overline{rs} \textbf{ = } \Omega[y]\ \textbf{in}\ (\textbf{let}\ \overline{\hat{rs}'} \textbf{ = } \hat{e}'_{body}\ \textbf{in}\ (reset, \overline{\hat{rs}'}, \overline{\hat{fv}})) \qquad \widehat{init} = (reset, \Lambda_1[\overline{lres}], \Lambda_1[\overline{fv}]) \\
\hat{e}_{loop} = \textbf{loop}\ (\hat{\overline{x}}, \overline{\hat{rs}}, \overline{\hat{fv}}) \textbf{ = } \widehat{init}\ \textbf{for}\ y = e_n - 1\ \textbf{to}\ 0\ \textbf{do}\ \hat{e}_{body} \qquad \Lambda_3 = \Lambda_1, \overline{fv} \mapsto \overline{\hat{fv}}
\end{array}}{\Lambda_1 \vdash e_{loop} \Lleftarrow \left(\Lambda_3 \vdash \textbf{let}\ (\overline{\hat{x}'}, \overline{\hat{rs}'}, \overline{\hat{fv}'}) \textbf{ = } \hat{e}_{loop}\ \textbf{in}\ (\textbf{let}\ \overline{\hat{fv}'} \textbf{ = } \overline{\hat{fv}} + \overline{\hat{rs}'}\ \textbf{in}\ \overline{\hat{fv}'})\right)}\ \textsc{RevLoop}
$$

$$
\frac{\begin{array}{c}
\Lambda \vdash e_t \Lleftarrow \Lambda_t \vdash \textbf{let}\ \overline{\hat{fv}_t} \textbf{ = } \hat{e}_t\ \textbf{in}\ \overline{\hat{fv}_t} \\
\Lambda \vdash e_f \Lleftarrow \Lambda_f \vdash \textbf{let}\ \overline{\hat{fv}_f} \textbf{ = } \hat{e}_f\ \textbf{in}\ \overline{\hat{fv}_f} \qquad \Lambda_{\Delta_t} = \Lambda \setminus \Lambda_t \qquad \Lambda_{\Delta_f} = \Lambda \setminus \Lambda_f \\
\overline{\hat{fv}}\ fresh \qquad \hat{e}'_t = \textbf{let}\ \overline{\hat{fv}} \textbf{ = } sort(\overline{\hat{fv}_t}\ +\!\!+\ im(\Lambda_{\Delta_f} - \Lambda_{\Delta_t}))\ \textbf{in}\ (\textbf{let}\ \overline{\hat{fv}_t} \textbf{ = } \hat{e}_t\ \textbf{in}\ \overline{\hat{fv}_t}) \\
\hat{e}'_f = \textbf{let}\ \overline{\hat{fv}} \textbf{ = } sort(\overline{\hat{fv}_f}\ +\!\!+\ im(\Lambda_{\Delta_t} - \Lambda_{\Delta_f}))\ \textbf{in}\ (\textbf{let}\ \overline{\hat{fv}_f} \textbf{ = } \hat{e}_f\ \textbf{in}\ \overline{\hat{fv}_f}) \\
\Lambda' = \Lambda,\ dom(\Lambda_{\Delta_t} \cup \Lambda_{\Delta_f}) \mapsto \overline{\hat{fv}}
\end{array}}{\Lambda \vdash \textbf{if}\ e_p\ \textbf{then}\ e_t\ \textbf{else}\ e_f \Lleftarrow \Lambda' \vdash \textbf{if}\ e_p\ \textbf{then}\ \hat{e}'_t\ \textbf{else}\ \hat{e}'_f}\ \textsc{RevIFE}
$$