$$
\begin{array}{lcl}
FV : \texttt{S.t} \to \texttt{SparcAsm.t} \to \texttt{S.t} & & \\
FV_s(x \leftarrow e; E) & = & s' = FV_s(E) \setminus \{x\} \text{ として } FV_{s'}(e) \\
FV_s(e) & = & FV_s(e) \\
 & & \\
FV : \texttt{S.t} \to \texttt{SparcAsm.exp} \to \texttt{S.t} & & \\
FV_s(c) & = & s \\
FV_s(\texttt{L}_x) & = & s \\
FV_s(op(x_1, \ldots, x_n)) & = & \{x_1, \ldots, x_n\} \cup s \\
FV_s(\texttt{if } x = y \texttt{ then } E_1 \texttt{ else } E_2) & = & \{x, y\} \cup FV_s(E_1) \cup FV_s(E_2) \\
FV_s(\texttt{if } x \leq y \texttt{ then } E_1 \texttt{ else } E_2) & = & \{x, y\} \cup FV_s(E_1) \cup FV_s(E_2) \\
FV_s(x) & = & \{x\} \cup s \\
FV_s(apply\_closure(x, y_1, \ldots, y_n)) & = & \{x, y_1, \ldots, y_n\} \cup s \\
FV_s(apply\_direct(\texttt{L}_x, y_1, \ldots, y_n)) & = & \{y_1, \ldots, y_n\} \cup s \\
FV_s(x.(y)) & = & \{x, y\} \cup s \\
FV_s(x.(y) \leftarrow z) & = & \{x, y, z\} \cup s \\
FV_s(\texttt{save}(x, y)) & = & \{x\} \cup s \\
FV_s(\texttt{restore}(y)) & = & s
\end{array}
$$

図 1: 命令の列 E および式 e において生きている変数の集合 $FV_s(E)$ および $FV_s(e)$。s は E や e の後で使われる変数の集合。以後の $FV(E)$ は $FV_\emptyset(E)$ の略記。

$\mathcal{R}:\texttt{SparcAsm.prog} \to \texttt{SparcAsm.prog}$

$\mathcal{R}((\{D_1,\ldots,D_n\},E)) \quad = \quad (\{\mathcal{R}(D_1),\ldots,\mathcal{R}(D_n)\},\mathcal{R}_\emptyset(E,x,())) \qquad x$ はダミーの fresh な変数

$\mathcal{R}:\texttt{SparcAsm.fundef} \to \texttt{SparcAsm.fundef}$

$\mathcal{R}(\texttt{L}_x(y_1,\ldots,y_n)=E) \quad = \quad \texttt{L}_x(\texttt{R}_1,\ldots,\texttt{R}_n)=\mathcal{R}_{x\mapsto\texttt{R}_0,y_1\mapsto\texttt{R}_1,\ldots,y_n\mapsto\texttt{R}_n}(E,\texttt{R}_0,\texttt{R}_0)$

$\mathcal{R}:\texttt{Id.t M.t} \to \texttt{SparcAsm.t} \times \texttt{Id.t} \times \texttt{SparcAsm.t} \to \texttt{SparcAsm.t} \times \texttt{Id.t M.t}$

$$\begin{array}{lcll}
\mathcal{R}_\varepsilon((x\leftarrow e;E),z_{\texttt{dest}},E_{\texttt{cont}}) & = & E'_{\texttt{cont}}=(z_{\texttt{dest}}\leftarrow E;E_{\texttt{cont}}), & \\
 & & \mathcal{R}_\varepsilon(e,x,E'_{\texttt{cont}})=(E',\varepsilon'), & \\
 & & r\notin\{\varepsilon'(y)\mid y\in FV(E'_{\texttt{cont}})\}, & \\
 & & \mathcal{R}_{\varepsilon',x\mapsto r}(E,z_{\texttt{dest}},E_{\texttt{cont}})=(E'',\varepsilon'')\ \text{として} & \\
 & & ((r\leftarrow E';E''),\varepsilon'') & x\ \text{がレジスタでない場合} \\
\mathcal{R}_\varepsilon((r\leftarrow e;E),z_{\texttt{dest}},E_{\texttt{cont}}) & = & E'_{\texttt{cont}}=(z_{\texttt{dest}}\leftarrow E;E_{\texttt{cont}}), & \\
 & & \mathcal{R}_\varepsilon(e,r,E'_{\texttt{cont}})=(E',\varepsilon'), & \\
 & & \mathcal{R}_{\varepsilon'}(E,z_{\texttt{dest}},E_{\texttt{cont}})=(E'',\varepsilon'')\ \text{として} & \\
 & & ((r\leftarrow E';E''),\varepsilon'') & \\
\mathcal{R}_\varepsilon(e,x,E_{\texttt{cont}}) & = & \mathcal{R}_\varepsilon(e,x,E_{\texttt{cont}}) & \text{(次図参照)}
\end{array}$$

図 2: 単純なレジスタ割り当て $\mathcal{R}(P)$, $\mathcal{R}(D)$ および $\mathcal{R}_\varepsilon(E,z_{\texttt{dest}},E_{\texttt{cont}})$。$\varepsilon$ は変数からレジスタへの写像、$z_{\texttt{dest}}$ は E の結果をセットする変数、$E_{\texttt{cont}}$ は E の後に実行される命令の列。$\mathcal{R}_\varepsilon(E,x,E_{\texttt{cont}})$ の返り値はレジスタ割り当てされた命令の列 E' と、E の後のレジスタ割り当てを表す写像 ε' の組。[ファイル `regAlloc.notarget-nospill.ml` 参照]

$$
\begin{array}{lcl}
\multicolumn{3}{l}{\mathcal{R} : \texttt{Id.t M.t} \rightarrow \texttt{SparcAsm.exp} \times \texttt{Id.t} \times \texttt{SparcAsm.t} \rightarrow \texttt{SparcAsm.t} \times \texttt{Id.t M.t}} \\
\mathcal{R}_\varepsilon(c, z_{\texttt{dest}}, E_{\texttt{cont}}) & = & (c, \varepsilon) \\
\mathcal{R}_\varepsilon(\texttt{L}_x, z_{\texttt{dest}}, E_{\texttt{cont}}) & = & (\texttt{L}_x, \varepsilon) \\
\mathcal{R}_\varepsilon(op(x_1, \ldots, x_n), z_{\texttt{dest}}, E_{\texttt{cont}}) & = & (op(\varepsilon(x_1), \ldots, \varepsilon(x_n)), \varepsilon) \\
\mathcal{R}_\varepsilon(\texttt{if } x = y \texttt{ then } E_1 \texttt{ else } E_2, z_{\texttt{dest}}, E_{\texttt{cont}}) & = & \mathcal{R}_\varepsilon(E_1, z_{\texttt{dest}}, E_{\texttt{cont}}) = (E_1', \varepsilon_1), \\
 & & \mathcal{R}_\varepsilon(E_2, z_{\texttt{dest}}, E_{\texttt{cont}}) = (E_2', \varepsilon_2), \\
 & & \varepsilon' = \{z \mapsto r \mid \varepsilon_1(z) = \varepsilon_2(z) = r\}, \\
 & & \{z_1, \ldots, z_n\} = \\
 & & \quad (FV(E_{\texttt{cont}}) \setminus \{z_{\texttt{dest}}\} \setminus dom(\varepsilon')) \cap dom(\varepsilon) \text{ として} \\
 & & ((\texttt{save}(\varepsilon(z_1), z_1); \ldots; \texttt{save}(\varepsilon(z_n), z_n); \\
 & & \;\; \texttt{if } \varepsilon(x) \leq \varepsilon(y) \texttt{ then } E_1' \texttt{ else } E_2'), \varepsilon') \\
\mathcal{R}_\varepsilon(\texttt{if } x \leq y \texttt{ then } E_1 \texttt{ else } E_2, z_{\texttt{dest}}, E_{\texttt{cont}}) & = & \text{同様} \\
\mathcal{R}_\varepsilon(x, z_{\texttt{dest}}, E_{\texttt{cont}}) & = & (\varepsilon(x), \varepsilon) \\
\mathcal{R}_\varepsilon(apply\_closure(x, y_1, \ldots, y_n), z_{\texttt{dest}}, E_{\texttt{cont}}) & = & \{z_1, \ldots, z_n\} = (FV(E_{\texttt{cont}}) \setminus \{z_{\texttt{dest}}\}) \cap dom(\varepsilon) \text{ として} \\
 & & ((\texttt{save}(\varepsilon(z_1), z_1); \ldots; \texttt{save}(\varepsilon(z_n), z_n); \\
 & & \;\; apply\_closure(\varepsilon(x), \varepsilon(y_1), \ldots, \varepsilon(y_n))), \emptyset) \\
\mathcal{R}_\varepsilon(apply\_direct(\texttt{L}_x, y_1, \ldots, y_n), z_{\texttt{dest}}, E_{\texttt{cont}}) & = & \text{同様} \\
\mathcal{R}_\varepsilon(x.(y), z_{\texttt{dest}}, E_{\texttt{cont}}) & = & (\varepsilon(x).(\varepsilon(y)), \varepsilon) \\
\mathcal{R}_\varepsilon(x.(y) \leftarrow z, z_{\texttt{dest}}, E_{\texttt{cont}}) & = & (\varepsilon(x).(\varepsilon(y)) \leftarrow \varepsilon(z), \varepsilon) \\
\mathcal{R}_\varepsilon(\texttt{save}(x, y), z_{\texttt{dest}}, E_{\texttt{cont}}) & = & (\texttt{save}(\varepsilon(x), y), \varepsilon) \\
\mathcal{R}_\varepsilon(\texttt{restore}(y), z_{\texttt{dest}}, E_{\texttt{cont}}) & = & (\texttt{restore}(y), \varepsilon)
\end{array}
$$

図 3: 単純なレジスタ割り当て $\mathcal{R}_\varepsilon(e, z_{\texttt{dest}}, E_{\texttt{cont}})$。$\mathcal{R}_\varepsilon(e)$ の右辺で変数 x のレジスタ $\varepsilon(x)$ が定義されていない場合は、$\mathcal{R}_\varepsilon(e) = \mathcal{R}_\varepsilon(x \leftarrow \texttt{restore}(x); e)$ とする。ただしレジスタ r については $\varepsilon(r) = r$ とする。[ファイル `regAlloc.notarget-nospill.ml` 参照]

$$
\begin{array}{lcl}
\mathcal{T} : \texttt{Id.t} \to \texttt{SparcAsm.t} \times \texttt{Id.t} \to \texttt{bool} \times \texttt{S.t} & & \\
\mathcal{T}_x((y \leftarrow e; E), z_{\texttt{dest}}) & = & \mathcal{T}_x(e, y) = (c_1, s_1) \text{ として、もし } c_1 \text{ ならば } (\mathit{true}, s_1) \\
 & & \text{そうでなければ } \mathcal{T}_x(E, z_{\texttt{dest}}) = (c_2, s_2) \text{ として } (c_2, s_1 \cup s_2) \\
\mathcal{T}_x(e, z_{\texttt{dest}}) & = & \mathcal{T}_x(e, z_{\texttt{dest}}) \\
 & & \\
\mathcal{T} : \texttt{Id.t} \to \texttt{SparcAsm.exp} \times \texttt{Id.t} \to \texttt{bool} \times \texttt{S.t} & & \\
\mathcal{T}_x(x, z_{\texttt{dest}}) & = & (\mathit{false}, \{z_{\texttt{dest}}\}) \\
\mathcal{T}_x(\texttt{if } y = z \texttt{ then } E_1 \texttt{ else } E_2, z_{\texttt{dest}}) & = & \mathcal{T}_x(E_1, z_{\texttt{dest}}) = (c_1, s_1), \\
 & & \mathcal{T}_x(E_2, z_{\texttt{dest}}) = (c_2, s_2) \text{ として} \\
 & & (c_1 \wedge c_2, s_1 \cup s_2) \\
\mathcal{T}_x(\texttt{if } y \leq z \texttt{ then } E_1 \texttt{ else } E_2, z_{\texttt{dest}}) & = & \text{同上} \\
\mathcal{T}_x(\mathit{apply\_closure}(y_0, y_1, \ldots, y_n), z_{\texttt{dest}}) & = & (\mathit{true}, \{\texttt{R}_i \mid x = y_i\}) \\
\mathcal{T}_x(\mathit{apply\_direct}(\texttt{L}_y, y_1, \ldots, y_n), z_{\texttt{dest}}) & = & \text{同上} \\
\mathcal{T}_x(e, z_{\texttt{dest}}) & = & (\mathit{false}, \emptyset) \qquad \text{それ以外の場合}
\end{array}
$$

図 4: 変数 x に割り当てるレジスタ r を選ぶときに使う targeting $\mathcal{T}_x(E, z_{\texttt{dest}})$ および $\mathcal{T}_x(e, z_{\texttt{dest}})$。$E$ や e で関数呼び出しがあったかどうかを表す論理値 c と、x を割り当てると良いレジスタの集合 s の組を返す。前々図の「x がレジスタでない場合」において、$\mathcal{T}_x(E'_{\texttt{cont}}, z_{\texttt{dest}}) = (c, s)$ として、できれば $r \in s$ とする。[ファイル `regAlloc.target-nospill.ml` 参照]

$$
\begin{array}{lcl}
\mathcal{R} : \texttt{Id.t M.t} \to \texttt{SparcAsm.t} \times \texttt{Id.t} \times \texttt{SparcAsm.t} \to \texttt{SparcAsm.t} \times \texttt{Id.t M.t} & & \\
\mathcal{R}_\varepsilon((x \leftarrow e; E), z_{\texttt{dest}}, E_{\texttt{cont}}) & = & E'_{\texttt{cont}} = (z_{\texttt{dest}} \leftarrow E; E_{\texttt{cont}}), \\
 & & \mathcal{R}_\varepsilon(e, x, E'_{\texttt{cont}}) = (E', \varepsilon'), \\
 & & y \in FV(E'_{\texttt{cont}}), \\
 & & \mathcal{R}_{\varepsilon' \setminus \{y \mapsto \varepsilon'(y)\}, x \mapsto \varepsilon'(y)}(E, z_{\texttt{dest}}, E_{\texttt{cont}}) = (E'', \varepsilon'') \text{ として} \\
 & & ((\texttt{save}(\varepsilon(y), y), r \leftarrow E'; E''), \varepsilon'') \\
 & & \qquad x \text{ がレジスタでなく、} \\
 & & \qquad r \notin \{\varepsilon'(y) \mid y \in FV(E'_{\texttt{cont}})\} \text{ なる } r \text{ がない場合}
\end{array}
$$

図 5: spilling をするレジスタ割り当て $\mathcal{R}_\varepsilon(E, z_{\texttt{dest}}, E_{\texttt{cont}})$ [ファイル `regAlloc.target-latespill.ml` 参照]