

$$\begin{array}{l}
\mathcal{T}[\text{if } (e) \quad = \{ \\
\quad s_1 \quad \text{char } c = !!e; \\
\text{else} \quad \text{if } (c) \\
\quad s_2] \quad \quad \mathcal{T}[s_1] \\
\quad \quad \text{else} \\
\quad \quad \mathcal{T}[s_2] \\
\quad \quad \text{SAVE}(c); \\
\quad \quad \} \\
\end{array}
\qquad
\begin{array}{l}
\mathcal{R}[\text{if } (e) \quad = \{ \\
\quad s_1 \quad \text{char } c; \\
\text{else} \quad \text{RESTORE}(c); \\
\quad s_2] \quad \quad \mathcal{R}[s_1] \\
\quad \quad \text{if } (c) \\
\quad \quad \mathcal{R}[s_1] \\
\quad \quad \text{else} \\
\quad \quad \mathcal{R}[s_2] \\
\quad \quad \} \\
\end{array}$$

(c は fresh)

図 1: if(仮)

if 文は、分岐の処理を行う構文である。if 文に記述されている条件が真であるとき、then 節が実行され、偽であるとき、else 節が実行される。else 節の記述は任意であり、else 節を記述せず、if 文の条件が偽であるとき、何も実行されない。if 文が実行されたとき、if 文の条件の真理値は記録されない。if 文の条件の真理値が記録されなければ、if 文を逆実行したとき、どちらの節を実行してよいか分からない。したがって、if 文を可逆化するためには、if 文の条件の真理値を記録する必要がある。

順実行を行うための変換では、if 文の条件の真理値を記録するための fresh な変数を用意している。更に、条件式を二重否定することで真理値へと変換し、fresh な変数とその真理値で初期化することで、if 文の条件の真理値を記録できる。また、fresh な変数には if 文の条件の真理値が記録されているので、if 文の条件の記述を fresh な変数へと書き換えている。逆実行を行うための変換では、fresh な変数に真理値を読み出し、変数を if 文の条件とすることで、順実行と同じ節を実行することができる。あとは、then 節と else 節に変換を再帰的に適用することで、if 文全体を変換できる。また、真理値は 1bit の情報であるため、fresh な変数を char 型で宣言することで、確保されるメモリを比較的小さいサイズに抑えている。