論文題目	命令型プログラミング言語におけるプログラム可逆化
研究目的	可逆計算は PDES のロールバックの高速化や双方向デバッガなどへの応用が知られている．Perumalla らの論文では，C 言語の構文範囲を限定し，その中で非可逆となる構文の制御情報や破棄される計算履歴などを外部記憶に保存することで C 言語の可逆化を行っている．しかし，Perumalla らの C 言語の可逆化は，逆実行を行った際に元の情報を復元できないような正しくない部分があり，また，外部記憶に保存する必要のない情報まで保存しているような効率的でない部分がある．これらを改善し，C 言語の可逆化を正しく行えるようになると，正しくない可逆化による予期せぬエラーの発生やバグの発生を防ぐことができる．また，C 言語の可逆化を効率的に行えるようになると，外部記憶や可逆化後のプログラムの空間計算量やオーバヘッドを減らすことができる．本研究では Perumalla らの C 言語の可逆化の定義を利用し，それらを正しく，効率的に可逆化できるように再定義を行うことを目的とする．
研究計画	Perumalla らの論文を用い，C 言語の可逆化の各定義に対して条件の指定と効率化できる定義の特定，効率化を行う．C 言語の可逆化を正しく行えるようにするために，論文で定義されている可逆化の条件を指定する．制御情報は変数を用いてプログラム内に定義され，その後，値が外部記憶に保存されるため，演算によって桁あふれなどのエラーが起こり，正常な値が保存されない場合がある．また，副作用をもつ演算では，逆実行によって元の値を復元できない場合がある．したがって条件は，制御情報の記憶に使われる変数の型，副作用をもつ演算子の種類などをもとに指定する．C 言語の可逆化を効率的に行えるようにするために，情報が外部記憶に保存されるように可逆化を行っている定義を見つける．可逆化後のプログラムは順実行のために元のプログラムの情報を保持しなければならない．したがって，可逆化前よりも情報を減らすことはできない．しかし，外部記憶に保存される情報は余分な情報であるため，条件を指定することで保存しないでよい場合がある．さらに，外部記憶への保存を行わないことで，オーバヘッドを減らすことができる．また，制御情報を保存する変数は，同じ空間計算量をもつ型の中で最適でない場合がある．したがって，効率化が可能だった場合，効率化の条件を指定して新たな可逆化を定義する．または，制御情報の型を変更する．中間審査までに Perumalla らの C 言語の可逆化の定義に対して条件付け，効率化を行い，その後，元の定義との比較を行うことで新しい定義を評価する．