

研究テーマ：C言語の可逆コンパイラ的设计と実装

背景：プログラムが可逆であるとき、そのプログラムは高々1つの前の状態をもつ。可逆プログラムはデバッグや並列シミュレーションなどに応用できることが知られており、実際に応用例が示されている。しかし、ほとんどのプログラミング言語は可逆性を前提として設計されていない。したがって、現実のほとんどのプログラムは可逆性を持たず、そのままではこれらの応用を適用することはできない。また、非可逆プログラムを可逆プログラムに変換する一般解法も知られているが、手作業による変換ではコストが大きく、人的なミスによりバグが引き起こされる可能性がある。

目的：本研究では、以前に提案された手法[1]を元に、C言語のコア部分の非可逆プログラムをC言語の可逆プログラムへと変換する自動変換器の設計と実装を行う。これにより、変換されたプログラムが可逆性を持つこと、自動変換による開発コストの削減が期待される。

可逆性の定義：状態が1つ前に戻れる。

可逆プログラムの定義：プログラムを実行した後、そのプログラムを逆実行することにより、変数の値を初期状態に戻すことができる。

状態：プログラム中の変数の値の組である。（後で形式的にも書く）

先週からの進捗状況：

1. [whileの変換の説明](#)にオーバーフローの議論を追加
2. [ifの変換の説明](#)を記述
3. [文の接続の変換の説明](#)を記述

今後の計画：

1. T[], R[]から変換器の作成
2. 変換器に関する説明を書く
3. テストによる正しさの証明（テストケースの作成）
4. Perumalla の乱数生成[2]をテストケースにする
5. 参考文献のまとめ
6. OCamlのまとめ
7. BNFでC言語のコア言語を定める

来週までにやること

参考文献について、不完全でいいのでテンプレを参考にまとめる

[1] Perumalla, K.: Introduction to Reversible Computing, Chapman and Hall/CRC (2013).

[2] Yognath, S.B. and Perumalla, K.S.: Efficient reversible uniform and non-uniform random number generation in UNU.RAN, *Proc. Annual Simulation Symposium (ANSS 2018)*,

2:1–2:10, Society for Computer Simulation International (2018). Available from <https://dl.acm.org/citation.cfm?id=3213034>