

研究テーマ：C言語の可逆化変換・逆可逆化変換の定式化

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一つの前の状態と一つ後の状態を持つ。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。そのため、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。

可逆プログラムの定義：

- ・メモリが1つ前の状態に戻る
- ・

先週からの進捗状況：

1. 変換器の開発状況の把握  
    テストケースの作成
2. 新しい変換規則の作成

今後の計画：

1. T[], R[]のサブセットから変換器の作成
2. サブセットの文章を書く
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

来週までにやること

1. whileの場合の実装
2. whileの変換の説明
3. 参考文献について、不完全でも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] Srikanth B. Yoginath, Kalyan S. Perumalla: Efficient reversible uniform and non-uniform random number generation in UNU.RAN. SpringSim (ANSS) 2018: 2:1-2:10