

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：【研究の背景を書く】

- ・可逆計算
 - ・逆実行ができる
 - ・デバッグやロールバックで有用
 - ・ほとんどのプログラミング言語は可逆でない

可逆性とは、ある状態の一つ後の状態と、一つ前の状態が高々一つしかない性質である。可逆性を持ったものはその性質から、状態を逆の順序で辿ることができる。

目的：【研究の目的を書く】

- ・C言語のコードを可逆化すること
 - ・文献[1]の変換の定式化
 - ・C言語のコードを元に順実行、逆実行のC言語のコードを自動生成

先週からの進捗状況：

1. 可逆化変換の変換規則の定式化

文献[1]のp162-166までの以下の可逆化変換の変換規則の定式化を行った。

1. ラベル付き文
2. 宣言
3. if文(else付き)
4. switch文

課題：

1. 適切な定式化は何か？

前回の助言「MinCamlの擬似コードを参考にすること」

<http://esumii.github.io/min-caml/min-caml.pdf>

今後の計画：

1. 研究計画を立てる
2. 参考文献を読んでまとめる【以下の内容をまとめる】
 - a. 研究分野
 - b. 目的
 - c. 背景
 - d. アプローチ
 - e. 結果
 - f. 有用性
 - g. 限界・短所
 - h. 次に何を読めばいいか？
3. 擬似コードの抽象構文木の定式化 7/12まで
4. 文献[1]に記載のすべての可逆化変換の定式化
5. 文献[1]に記載のすべての逆可逆化変換の定式化
6. 「令和元年度 電気・電子・情報関係学会 東海支部連合大会」への投稿
論文投稿（7月16日（火）23:00 ㄨ）
7. OCamlのまとめ

[1] Perumalla, K.: Introduction to Reversible Computing, Chapman and Hall/CRC, (2013).
情報処理学会の「論文誌ジャーナル (IP SJ Journal) 原稿執筆案内」を参考にして参考文献
リストを作る. https://www.ipsj.or.jp/journal/submit/ronbun_j_prms.html