

～変える～

研究テーマ：命令型プログラミング言語におけるプログラム可逆化

背景：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ーヘッドを減らすことができる。

目的：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。

比較はPerumallaの手法と行う

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）

～ここまで～

先週からの進捗状況：

1. [単純な例について調べた](#)
2. ループの例を文章化した（上と同じリンク内）
3. C言語によるアルゴリズム事典で静的な算術符号が有効なアルゴリズムを調べている

次回までに進めること

- 1.