定例進捗報告 2020年10月13日(火) 15:15～

修士2年 M2019SE009 渡邉将匡

～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. 算術符号をループの制御情報の圧縮に使った場合，どれだけのビット数が必要になるか勉強した（log2(n+1)のシーリングだった）
2. 制御情報の圧縮が有効そうなアルゴリズムを探した（探索，ソート）

・算術符号
  ・ループの制御情報を圧縮
  ・ループがn回繰り返す場合
    ・つまり，記号が1と0で頻度がn/n+1と1/n+1の場合
      ・ループが続くが1でループが続かないが0
  ・算術符号の平均符号長はエントロピーと同じ
  ・よって(1/n+1)*log2(n+1)が平均符号長
    ・-(n/n+1)log2(n/n+1) - (1/n+1)*log2(1/n+1)

・-(n/n+1)log2(n/n+1)はほぼ0になる
・そこに記号列の長さのn+1を掛けてlog2(n+1)のシーリングが必要ビット数
・ビット列で表す場合，n+1ビット必要なので確かに圧縮できている
・繰り返し回数で表す場合，log2(n+1)のシーリングが必要ビット数なので，ループの制御情報はこれ以上圧縮できない

・算術符号が有効なアルゴリズム
・探索
・特に線形探索などの，キーと一致するか比較する以外にif文がないパターン
・ソート
・値の入れ替えと入れ替えない操作の確率が半々とならない場合

次回までに進めること
1.