

TIME/SPACE TRADE-OFFS FOR REVERSIBLE COMPUTATION

1. 研究分野

可逆コンピューティング
ー可逆アルゴリズム

2. 目的

既存の方法(Bennett法など)などよりも空間使用量の観点から効率的な可逆ミュレーションを提案する.

3. 背景

既存の方法(Bennett法など)は, 空間使用量が効率的ではなく, 場合によっては非可逆な場合に比べて指数関数的な増加が見られた.

Bennett法

時間計算量 T , 空間計算量 S の非可逆アルゴリズムは, 時間計算量 $O(T)$, 空間計算量 $O(S+T)$ で可逆シミュレーションできる[C. H. Bennett, *Logical reversibility of computation*].

→場合によっては, 空間使用量はシミュレーション前に比べて指数関数的に増大.

4. アプローチ

入力保存可逆シミュレーション

時間計算量 T , 空間計算量 S の非可逆アルゴリズムは, 時間計算量 $O(T^{\{1+\epsilon\}})$, 空間計算量 $O(S \log T)$ で可逆シミュレーションできる. * $\epsilon=(1/\log k)>0$

方法:

1. 非可逆アルゴリズムを $k>1$ ステップごとのセグメントとして分割する.
2. k ステップ計算が進む毎に, Bennett法を用いて履歴を消去する

計算量:

- ・ 計算は $(2k-1)$ ステージで行われる.
- ・ それぞれのステージでは, 計算は線形時間で行われる.
- ・ 各ステージの計算途中では, $(k-1)$ のチェックポイントが現れる.
- ・ 各チェックポイントに必要な空間計算量は $O(S)$ である.
- ・ k の値は $O(\log T)$ に収まる.
- ・ よって必要な空間計算量は, $k * \text{チェックポイント数} = O(S \log T)$ である.
- ・ 各ステージに必要な時間計算量は $O(S)$ である.
- ・ よって必要な時間計算量は, $S * \text{ステージ数} = O(S(2k-1)) = O(T^{\{1+(1/\log k)\}})$ である.

入力消去可逆シミュレーション

入力を消去するような可逆シミュレーションは次の方法で行うことができる。また、この方法では、計算に用いる関数 f が単射であるか非単射であるかによって計算時間が異なる。ここで単射関数について、関数 f の逆を計算する関数 f^{-1} は、時間計算量 $O(T^*)$ 及び空間計算量 $O(S^*)$ で計算可能である。

方法：

1. 入力保存可逆シミュレーションの要領で、入力及び出力を保持した状態を用意する
2. 計算に用いた関数 f の逆を計算する関数 f^{-1} を用意する。
3. f^{-1} を用いて入力を消去する。

計算量：

- ・ f が単射関数の場合
 - 時間計算量 $O(T+T^*)$
 - 空間計算量 $O(\max\{S^*T^\epsilon, S^{**}T^{*\epsilon}\})$
- ・ f が非単射な場合
 - 時間計算量 $O(T^{1+\epsilon}+T^{*1+\epsilon})$
 - 空間計算量 $O(\max\{S\log T, S^*\log T^*\})$

5. 結果

この論文では、著者Bennettが以前提案した方法について、より空間的に効率的な方法を与えた。この方法では、時間計算量にオーバーヘッドが存在するが空間的な効率性は大きい。

6. 有用性

空間使用量はアルゴリズムの性能を考える上で重要である。

7. 限界・短所

8. 次に何を読めばいいか？

Bennett法について

- ・ C. H. BENNETT, Logical reversibility of computation, IBM J. Res. Develop., 17 (1973), pp. 525-532.