

# 自己調整二分木の並列操作

上田 和紀

Sleator と Tarjan による自己調整二分木 (splay tree) に対して並列操作を可能にする操作アルゴリズムを提案する。提案するアルゴリズムは、同一の木に対する複数の更新・挿入・削除操作のパイプライン的並列実行を許し、かつ操作系列のスループット (単位時間内に処理可能な操作の個数) とレスポンス (個々の操作の償却計算量 (amortized complexity)) を両立させることを目的としている。スループットの最適性と挿入操作の対数的レスポンスについては理論的結果を示す。削除操作は、木の形状に関する良い性質を保つにもかかわらず、Sleator らの枠組みでは最適性が証明できない。このことについても論じる。

## 1 はじめに

自己調整二分木 (スプレー木, splay tree) [2] は、アクセスした節点に対して扁平化 (splaying) 操作 (2.1 節) を施すことにより、木の形状を動的に最適化する二分探索木の総称であり、多くの強力な性質が成り立つことがわかっている。本論文では、同一のスプレー木に対する複数の挿入削除等の操作のパイプライン的並列実行を可能にする方法を検討する。目標は、下記の要請を満たす操作アルゴリズムを得ることである。

1. (レスポンス) 通常のスプレー木の操作と同様、対数的な償却計算量 (amortized complexity) [4] をもつ。
2. (スループット) 操作後の木の形状が、根に近い部分から葉に向かって漸増的に確定するようにすることで、個々の操作が同時に施錠しなければならない節点の数を高々  $O(1)$  個におさえる。

もしスループットだけが目標ならば、二分木を用いなくても、線形リストを用いて容易に達成できる。したがって、レスポンスとスループットを同時に達成する

ことが本質的に重要である。B 木やその変種に対する並列操作の研究は少なくない [1] が、スプレー木の並列性に関する研究は少なく、著者の知る限り、上記の二条件を満たす並列アルゴリズムはまだ提案されていない。

本論文では、二分探索木の各節点はキーと値の対を保持するものとし、節点はキーの対称順 (symmetric order) に並んでいるとする。基本操作として、次の二つを考える。単なる節点値の読み出しは *update* の単純な変種と考えることができる。

*update*( $i, v, v', t$ ): キー  $i$  をもつ節点が木  $t$  の中にあれば、その節点の現在の値を  $v$  に代入したあと、節点に新たな値  $v'$  を格納する。なければ、キー  $i$  と値  $v'$  をもつ節点を  $t$  に挿入し、 $v$  には節点がなかったことを示す特別の値を代入する。  
*delete*( $i, v, t$ ): キー  $i$  をもつ節点が木  $t$  の中にあれば、その節点の現在の値を  $v$  に代入したあと、節点を消去する。なければ  $v$  に特別の値を代入する。

## 2 関連研究

### 2.1 扁平化とトップダウン扁平化

スプレー木における扁平化とは、節点の探索操作においてアクセスしたパスの長さをおよそ半分にし

Concurrent Operations on Splay Trees.

Kazunori Ueda, 早稲田大学理工学部情報学科, Dept. of Information and Computer Science, Waseda University,