EFFICIENT LINKED LIST RANKING ALGORITHMS AND PARENTHESES MATCHING AS A NEW STRATEGY FOR PARALLEL ALGORITHM DESIGN

DISSERTATION

Presented to the Graduate Council of the University of North Texas in Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Ranette Hudson Halverson, B.S., M.S.
Denton, Texas
December, 1993

The goal of a parallel algorithm is to solve a single problem using multiple processors working together and to do so in an efficient manner. In this regard, there is a need to categorize strategies in order to solve broad classes of problems with similar structures and requirements. In this dissertation, two parallel algorithm design strategies are considered: linked list ranking and parentheses matching.

Deterministic and randomized linked list ranking algorithms are presented for the exclusive-read exclusive-write (EREW) parallel random access machine (PRAM) model. They are based on a technique unlike the traditional reduction method. The randomized algorithm is work-optimal, and, although the deterministic is not, the technique is quite simple in comparison to previously proposed algorithms and has the advantage of small constant factors in terms of time and space requirements.

Another contribution of this dissertation is the establishment of parentheses matching as a general strategy for designing efficient parallel algorithms. This is accomplished through the development of a class of tree related algorithms for the PRAM model which are solved using parentheses matching as a major component. The problems solved include the heights and extreme values of the nodes of a tree, the least common ancestor problem, inorder traversal of a tree, tree contraction, and balancing binary trees. Finally, a hypercube implementation for parentheses matching and its application to the nearest enclosing parentheses problem are presented.
ACKNOWLEDGEMENTS

I wish to thank my advisor, Dr. Sajal K. Das, for the benefit of his expertise and wisdom, which he so generously shared with me. I will be forever grateful for the experience I obtained while working with him.

I want to thank the members of my Ph.D. committee, Dr. Roy T. Jacob, Dr. C. Q. Yang, and Dr. Neal Brand for their time and effort on my behalf. I also thank the faculty of the UNT Department of Computer Science for their teaching, which helped prepare me for this dissertation. Thanks also to my friend and colleague, Richard Simpson, who never failed to encourage me when times were rough.

Finally, I wish to thank my family. Without their continual love and support this dissertation would not have been possible. I am grateful to my parents, Rita and Randall Hudson, who instilled in me the value of an education and a job well-done; to my husband, Patrick, who demonstrated the patience and support that few people possess; and to my sons, Stuart and Andrew, who make the future important.

"I can do all things through Christ Jesus who strengthens me." Philippians 4:13.
# TABLE OF CONTENTS

1 INTRODUCTION ........................................ 1
   1.1 Motivation ........................................ 1
   1.2 Performance Metrics ............................... 2
   1.3 Models of Computation ............................. 3
       1.3.1 PRAM Model .................................. 4
       1.3.2 Hypercube Model .............................. 5
   1.4 Parallel Algorithm Design Strategies ............... 6
   1.5 Research Overview ................................ 10

2 EXISTING WORK ON LINKED LIST RANKING ............... 12
   2.1 Introduction ...................................... 12
   2.2 Problem Definition ................................ 12
   2.3 Evolution of Parallel List Ranking Algorithms .... 14
   2.4 Parallel List Ranking Approaches ................ 17
       2.4.1 Pointer Jumping (Doubling) Approach ....... 18
       2.4.2 Reduce–Rank–Expand Technique .............. 21
   2.5 Conclusion ........................................ 27

3 LINKED LIST RANKING ALGORITHMS ON EREW PRAM .... 29
   3.1 Introduction ...................................... 29
   3.2 APRAM Model and Performance Metrics ............... 29
   3.3 APRAM List Ranking Algorithm .................... 30
   3.4 EREW List Ranking Algorithms .................... 32
3.4.1 Randomized Algorithm ........................................... 33
3.4.2 Deterministic Algorithm ........................................ 36
3.5 Advantages of New Algorithms .................................. 40
3.6 Conclusion .......................................................... 42

4 PARENTHESES MATCHING AS A DESIGN STRATEGY ............ 43
4.1 Introduction ............................................................ 43
4.2 Existing Work with Parentheses Matching ...................... 45
4.3 Existing Work in the Application of Parentheses Matching .. 48
4.3.1 Breadth-first Traversal of a Tree .............................. 48
4.3.2 Sorting Integers in a Restricted Class ....................... 49
4.3.3 Coloring of an Interval Graph .................................. 49
4.4 Application of Parentheses Matching to Trees ................ 50
4.4.1 Representations and Traversals of Trees ..................... 50
4.4.2 Tree Representations Related to Parentheses ............... 53
4.4.3 Euler Tour Technique vs. Parentheses Matching ............ 54
4.5 Parentheses Matching Applied to Basic Tree Operations ....... 54
4.5.1 Tree Traversals via PPM ........................................ 55
4.5.2 Converting Parentheses to Trees ............................. 55
4.5.3 Constructing a Tree from Traversals ......................... 58
4.5.4 Tree Contraction by Parentheses Matching .................. 59
4.6 Conclusion ............................................................ 62

5 APPLICATION OF PPM TO TREE RELATED PROBLEMS ......... 64
5.1 Introduction ............................................................ 64
5.2 Bottom-up Tree Computations .................................... 64
5.2.1 Heights of all Nodes of a Tree ............................... 66
5.2.2 Extreme Values of the Subtrees in a Tree .................. 68
5.2.3 Nearest Enclosing Parentheses ............................ 68
5.2.4 Lowest Common Ancestor ................................. 70
5.3 Balancing Binary Trees ...................................... 72
  5.3.1 Definitions and Notations .............................. 72
  5.3.2 Existing Work ........................................ 73
  5.3.3 General Balancing Strategy ............................ 74
  5.3.4 Perfect Trees via PPM ................................ 75
  5.3.5 Complete Trees via PPM ............................... 80
5.4 Conclusion .................................................. 84

6 PARENTHESES MATCHING ON A HYPERCUBE ................. 85
  6.1 Introduction ............................................. 85
  6.2 Existing Work ........................................... 85
    6.2.1 Parentheses Matching through Routing ............... 86
    6.2.2 Parentheses Matching Using ANSV ................... 87
    6.2.3 Divide-and-Conquer Technique ...................... 88
    6.2.4 Example ........................................... 90
  6.3 Hypercube Implementation ............................... 90
    6.3.1 Algorithm 2-CUBE ................................ 92
    6.3.2 Algorithm 4-CUBE ................................ 92
    6.3.3 Hypercube Parentheses Matching Algorithm .......... 94
  6.4 Time Complexity Analysis ................................ 95
  6.5 Possible Speedups ....................................... 97
  6.6 Comparison of PPM Algorithms ............................ 98
  6.7 Nearest Enclosing Parentheses Problem .................. 99
LIST OF TABLES

2.1 Applications of Linked List Ranking ........................................ 15
2.2 Comparison of List Ranking Algorithms ................................. 17
3.1 Demonstration of the EREW Randomized Algorithm .................. 35
3.2 Demonstration of the EREW Deterministic Algorithm ................. 38
3.3 Space Requirements of List Ranking Algorithms ....................... 41
4.1 Tree Contraction by Parentheses Matching ............................... 62
5.1 Demonstration of Heights of all Nodes ................................... 67
5.2 Demonstration of Algorithm NEPA .......................................... 70
5.3 Tree Related Algorithms using PPM ....................................... 83
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Shared Memory PRAM Model</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>Hypercube Model for n=2,4,8</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Linked List in Array Form</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Linked List in Logical Form</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>Strategy for Work-Optimal List Ranking</td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Demonstration of Pointer Jumping</td>
<td>19</td>
</tr>
<tr>
<td>2.5</td>
<td>Pointer Jumping Method for List Ranking</td>
<td>21</td>
</tr>
<tr>
<td>2.6</td>
<td>Deleting an Element from a Linked List</td>
<td>24</td>
</tr>
<tr>
<td>2.7</td>
<td>List Ranking Example</td>
<td>24</td>
</tr>
<tr>
<td>2.8</td>
<td>Linked List Expansion</td>
<td>26</td>
</tr>
<tr>
<td>4.1</td>
<td>Parentheses Matching Strategy</td>
<td>44</td>
</tr>
<tr>
<td>4.2</td>
<td>Encoding of Parentheses</td>
<td>47</td>
</tr>
<tr>
<td>4.3</td>
<td>Sample Tree</td>
<td>52</td>
</tr>
<tr>
<td>4.4</td>
<td>Parentheses Representations of a Tree</td>
<td>53</td>
</tr>
<tr>
<td>4.5</td>
<td>Demonstration of Algorithm INORDER</td>
<td>56</td>
</tr>
<tr>
<td>4.6</td>
<td>Demonstration of Algorithm PARENTHESES-TO-TREE</td>
<td>58</td>
</tr>
<tr>
<td>5.1</td>
<td>Demonstration of Range Maxima Problem</td>
<td>66</td>
</tr>
<tr>
<td>5.2</td>
<td>Complete and Perfect Trees of Twelve Nodes</td>
<td>73</td>
</tr>
<tr>
<td>5.3</td>
<td>Balancing a Binary Search Tree</td>
<td>79</td>
</tr>
<tr>
<td>5.4</td>
<td>Embedded Parentheses for Balanced Binary Tree</td>
<td>79</td>
</tr>
<tr>
<td>5.5</td>
<td>Generation of Parentheses String</td>
<td>83</td>
</tr>
<tr>
<td>6.1</td>
<td>Divide-and-Conquer Parentheses Matching Technique</td>
<td>91</td>
</tr>
</tbody>
</table>
6.2 Algorithm 4-CUBE Distribution of Data ................. 94
6.3 Marking of Subsequences of Parentheses ............... 96