

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

PREVIEW

Regular Array Expansions in Null Arrays, with Applications:

Zero Is Not Nothing

by

Ronald I. Frank

**Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing**

at

School of Computer Science and Information Systems

Pace University

June 2002

PREVIEW

UMI Number: 3064835

Copyright 2002 by
Frank, Ronald I.

All rights reserved.

UMI[®]

UMI Microform 3064835

Copyright 2003 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© Ronald I. Frank 2002

All Rights Reserved

PREVIEW

We hereby certify that this dissertation, submitted by Ronald I. Frank, satisfies the dissertation requirements for the degree of Doctor of Professional Studies and has been approved.

Dr. Fred Grossman

Date

Chairperson of Dissertation Committee

Dr. Michael Gargano

Date

Dissertation Committee Member

Dr. Allen Stix

Date

Dissertation Committee Member

School of Computer Science and Information Systems

Pace University 2002

Abstract

I show how a simple observation about mapping the binomial expansion to regular arrays leads to a decomposition of N-D equilateral regular arrays (ERAs) of axis length n , in terms of ERAs of all dimensions from 0 to N but of axis length one less ($n-1$) or one more ($n+1$). I then recurse the decompositions and derive a single canonical expansion of an N-D ERA of axis length n , in terms of null arrays.

I recapitulate these arguments and generalize them for non-equilateral regular arrays (NERAs). I call the final canonical decomposition of a NERA in terms of only null arrays the “master equation” because all other results can be derived from it. I derive many examples and subsidiary results along the way including the expansions in nulls for an arbitrary inner product and for a catenation of two conformable arrays. I also have found explanations for some open array questions.

There are four chapters on applications: one suggests a new operation on arrays – array division, and one suggests analyzing hyper cube machines and generalized hyper cube machines using these methods. The third applies these methods to OLAP hierarchical data, and the last one suggests enlarging computer languages to properly allow null arrays.

This is a rich set of results, so I indicate a number of possible new avenues of research applying these results. This includes array valued polynomials, fractional and complex length arrays, “ragged” arrays, complex and fractional dimension arrays, and GHC connectivity .

Key Words

Null Array, Binomial Array Expansion, Processor Connectivity, N Dimensional Machines, Null Data Structures, Computer Language Array Indexing, Generalized Hyper cubes, Array Division, Catenation, Lamination, Ragged Arrays.

PREVIEW

Corrigendum (added in proof)

Negative array axis length results are inserted in this thesis as side comments and not as direct contributors to the main arguments. This is distinguished from negative arrays, which are main line items. The negative axis length results in the NERA case (equations 6.1.2 - 6.1.9, pp. 91-93) may be partially wrong or incomplete. Therefore for now, all negative axis length results should be considered preliminary and that entire topic as a whole should be considered part of Chapter 8, Implications for Future Research.

Preface and Acknowledgement

I have been developing the ideas in this thesis over many years in fits and starts. I wrote up a series of very preliminary chapters covering some of the decomposition material during the late 1980s and early 1990s as a personal time project while at IBM (I lucked out and got one of the first IBM PCs in a personal purchase lottery). I worked in the APL development group from 1970 to 1974 in Philadelphia, and in the APL R & D group at IBM Research from 1985 to 1987. This thesis work was done mostly after I left IBM in 1993. See the reference below to the 1987 IBM-APL-ITL (Interdivisional Technical Liaison) meeting.

To the best of my knowledge, these ideas are uniquely mine developed on my own time with my own resources. Although I received personal encouragement from Adin Falkoff, my manager at the time, my subsequent management was not so encouraging, as my required research was always set in other directions and there was no interest in theoretical data structures. I never got to work explicitly on this topic during the work days except to prepare an IBM ITL presentation while reporting to Adin Falkoff (see below). All of the work on this thesis was done long after I retired from IBM.

Many people have influenced me in general. I can't remember them all nor can I isolate out which attitudes of mine came from which person or from interactions of which group of people.

What does stand out in my mind are the lunches of the APL Development Group. These were informal meetings, held at lunch time around a large conference table in the IBM Philadelphia Scientific Center during 1970-1974. While we ate our lunch, we bandied around various ideas. Most topics were related to extensions of the APL language. Some were about applications of APL. Some were about more fundamental mathematics or array ideas. Some were about the politics and the sociology of APL or IBM product development. Others were about general politics. All were stimulating.

I also was influenced by the larger IBM APL community through the various IBM ITL (Interdivisional Technical Liaison) meetings. I did present a very preliminary version of the first level decomposition (of regular cubes) at one of these meetings in 1987.

I have also been influenced by the larger APL community via the International Conferences, some of which I did attend, and at which I did present papers on other topics.

When I was at the IBM Cambridge Scientific Center, I helped an MIT student (Jeremy Bloom) with his thesis that involved a substantial APL implementation. This brought me into contact with the MIT National Bureau of Economic Research. Their TROLL language was, as I remember it, a FORTRAN based language for econometric modeling that incorporated many callable routines that I viewed as essentially the APL array primitives under other names. This enlarged my views of array processing.

To name names is dangerous not only because I will forget some, but because many conversations in the APL or IBM Research communities were intellectually adversarial so that the person might not want to be associated with this work.

Nevertheless, I remember interacting with Yon Bard, Phil Benkhard, Jim Brown, Graham Driscoll, John Gerth, Adin Falcoff, Dave Grossman, Ken Iverson, Dick Lathwell, Dan Ling, Gene McDonald, John McGrew, Trenchard More Jr., Don Orth, Ray Polivka, Dave Rabenhorst, Cory Scutt, Howard Smith, Beth Tibbits, and many others whose names escape me now.

None of these interactions were directly about this work except for an informal presentation to Dan Ling about my personal work. He was not encouraging, but I did find the presentation useful in orienting me to better explain the uses of null arrays.

I had many conversations with Trenchard More Jr. while we were at the IBM Cambridge Scientific Center. We talked mostly about his emerging “Array Theory”. I think I helped in part by introducing the idea of pointer or address name as an array element [24]. The important point concerning this thesis is that none of our conversations were about these ideas or about purely structural approaches to arrays. As I see it now, this work is orthogonal to Array Theory, intersecting only at the “origin”, i.e. our use of the scalar as null array of 0 dimension.

More recently, at Pace University, my thesis advisor Fred Grossman annoyingly kept wanting more than a coherent, novel, well worked out theoretical framework. He kept asking “so what?”, meaning that new knowledge, in and of itself, is not enough. We disagree. However, to placate this old friend and keep him quiet, I found some applications for him. The applications discussed here are merely indications of major independent investigations that could be undertaken. I don’t consider the four application sections complete expositions, but merely indications of how my ideas can be profitably applied in other areas.

Fred and I come out of the NYU Courant Institute Of Mathematical Sciences - Göttingen applied mathematics tradition of David Hilbert and Richard Courant. I remember Lipman Bers pointing out that there are an infinity of true theorems but one should only work on those that have consequences. My theorems have many consequences.

I also remember reading the number theorist G. H. Hardy's proud comment in his Mathematicians Apology [15], that he never worked on any mathematics that had an application. So, this disagreement goes way back. Eugene Wigner [28] pointed out the "unreasonable effectiveness of mathematics" in physics. I think Truth will eventually find a use. Even Hardy has a central result in genetics (The Hardy-Weinberg Law) named after him, because he derived it, theoretically, in passing.

I must thank Michelle Lang, Anne Campbell and the staff of the Pace libraries for excellent support in retrieving most of the references I used here.

Special note should be made of the opportunities afforded to me by Dean Susan Merritt, Dean of the Pace University School of Computer Science and Information Systems. She was instrumental in my undertaking this entire program by making me an "offer I couldn't refuse" - it was so good.

In previous studies, years ago, that prepared me for this work, I received substantial help from Claire and Harold Leegant, my inlaws. During those studies I received encouragement from Prof. Peter Lax of the NYU Courant Institute.

Over the years, early on, I have to remember my parents and my Uncle Harry for encouragement and guidance. I also want to point out the many and various forms of help I got from my wife Dr. Judith M. Frank, including encouragement and editing. She didn't complain about the infinity of chores not done.

Table of Contents

ABSTRACT.....	IV
Key Words	v
CORRIGENDUM (ADDED IN PROOF)	VI
PREFACE AND ACKNOWLEDGEMENT.....	VII
TABLE OF CONTENTS	X
LIST OF FIGURES.....	XVI
LIST OF TABLES.....	XXI
LIST OF THE MOST IMPORTANT EQUATIONS (28 OF 278)....	XXII
CHAPTER 1. INTRODUCTION.....	1
1.1 Writing Style and Local Shorthand	1
1.2 Flow of Development	2
1.3 Methodology Overview.....	3
1.4 Comments on Notation	8
1.5 Previous Work and Current Relevance.....	9
1.5.1 Previous Work.....	9
1.5.2 Current Relevance.....	12
1.6 Informal Definitions of Array.....	14
CHAPTER 2. ERA THEOREMS 1, 1+, AND FORMAL DEFINITIONS	19
2.1 Formal Notations	20
2.1.1 Tuple, Index, Axis, Dimension, and Rank.....	20

2.1.2	<i>Index Origin 1, and 0 Index Value.</i>	21
2.1.3	<i>Null Array Notation by Changing the Usual Max/Min Index Vector Values.</i>	21
2.1.4	<i>Index [second meaning]</i>	21
2.1.5	<i>Shape, Vector, Shape Vector, Equilateral, and Length.</i>	21
2.1.6	<i>Equilateral Array, Cells, and Cubes.</i>	22
2.1.7	<i>Array Contents, Mapping, Domain, Set of Values, and Co-domain.</i>	22
2.1.8	<i>Array Pictures</i>	23
2.1.9	<i>Regular Arrays and Ragged Arrays</i>	24
2.1.10	<i>Null Arrays and Scalars.</i>	25
2.1.11	<i>Empty Arrays (this is the APL Terminology for our Null Arrays).</i>	26
2.1.12	<i>Scalar Size (there is none)</i>	26
2.1.13	<i>Partition and Family.</i>	27
2.2	Notational Correspondence: Counting Array Cells and Arrays	29
2.2.1	<i>Counting Array Cells (the T Notation).</i>	29
2.2.2	<i>Array Notation Examples</i>	29
2.2.3	<i>Counting Cells in Arrays of the Same Shape</i>	30
2.2.4	<i>Counting with Binomial Coefficients.</i>	30
2.2.5	<i>Some Null Array Notation Examples.</i>	30
2.3	Theorem 1 – First Level Equilateral Arrays	31
2.3.1	<i>First-Level Decomposition for Smaller Arrays.</i>	31
2.3.2	<i>Binomial Theorem Preliminary.</i>	32
2.3.3	<i>Comment and Notation.</i>	32
2.3.4	<i>Array Notation.</i>	32
2.4	Theorem 1. The ERA Expansion	33
2.4.1	<i>Counting Arrays of the Same Shape</i>	34
2.4.2	<i>Counting with Binomial Coefficients.</i>	34
2.5	Proof of the Partition P	36
2.5.1	<i>Comment on the Proof of (2.4.1) Theorem 1</i>	41
2.5.2	<i>Proof of the Partition F</i>	41
2.5.3	<i>(2x2) Example</i>	42
2.5.4	<i>(3x3x3) Example</i>	43
2.5.5	<i>A Comment on the Alternate Form of (2.4.1).</i>	44
2.5.6	<i>Another Equivalent Form of (2.4.1).</i>	45
2.5.7	<i>Atomicity.</i>	46
2.6	Some Immediate Corollaries of Theorem 1 and Expansion (2.4.1) OR (2.5.16)	47
2.6.1	<i>0-D Cubes are Scalars:</i>	47
2.6.2	<i>Scalars are Atomic: (Uses (2.5.16) result. See Theorem 1+ Section 7)</i>	47
2.6.3	<i>1-D Cubes (Vectors):</i>	47
2.6.4	<i>Null Vectors (1D Null Cubes) are Atomic:</i>	48

2.6.5 Negative Length 0-D Cubes are Scalars:.....	48
2.6.6 Negative Length Vectors:.....	48
2.6.7 N-D Single Cell Arrays are a Sum of Nulls:.....	49
2.6.8 N-D Negative Single Cell Arrays are an Alternating Sum of Nulls:	49
2.6.9 Negative-D Null Cubes are UNDEFINED:.....	49
2.6.10 Negative Length N-D Cubes are $(-1)^N$ x (N-D Cubes):.....	50
2.6.11 Null Cubes are Atomic: $T_0^N = T_0^N$ a Heuristic Argument for the Ansatz.	53
2.7 Theorem 1+ - First Level Equilateral Arrays (n+1)	54
2.7.1 Theorem 1+	54
2.7.2 An (n+1) Example: The 3x3x3 Array	56
CHAPTER 3. THE FULL ERA EXPANSION THEOREM 2	58
3.1 Triangular Sums - Working up to the Proof.....	58
3.2 Theorem 2 Full ERA Decomposition	64
3.1.1 Null Cube Arrays are Atomic	65
3.1.2 Example of Some Expansion Coefficient Sums	65
3.3 An Updated Table of ERA Results.....	67
CHAPTER 4. FIRST LEVEL NERA EXPANSION THEOREM 1'	68
4.1 First-level NERA Decomposition for Smaller Arrays, P(A), and F^i	68
4.2 Binomial-Like Theorem Preliminary.....	69
4.2.1 Combinatorial Formula [8].....	69
4.2.2 The Generalization of the Binomial Expansion.....	70
4.2.3 The Array Notation for the General NERA.....	70
4.3 Theorem 1' First Level NERA Decomposition (4.3.1).....	71
4.3.1 A 2x2x3 NERA Example	72
4.4 Theorem 1+ '	74
4.4.1 Derivation of (2.7.1) from (4.4.1)	75
4.5 Corollaries of Theorem 1' (Transposition and Outer Product)	75
4.5.1 Transposition.....	75
4.5.2 Outer Product.....	75
4.5.3 Theorem: Outer Product Decompositions	78
4.5.4 Application of Theorem 1+ ' : All n_i Negative.....	80
4.5.5 Example: Outer Product of Scalars	80

4.5.6 Example: Outer Product of a Scalar and a Vector.....	80
CHAPTER 5. THE FULL NERA EXPANSION THEOREM 2'	81
5.1 Theorem 2' (5.1.1).....	81
5.2 Theorem 1+' NERA Decomposition in Larger Arrays Revisited.....	90
CHAPTER 6. THE MASTER EQUATION AND COROLLARIES....	91
6.1 The Master Equation.....	91
6.1.1 One n_i Negative	91
6.1.2 The Meaning of the Master Equation Illustrated by A (2x3x4) Example.	93
6.1.3 Duality of Representation.	95
6.1.4 Odometering.....	100
6.1.5 The (2x2x2) Sub Array example.....	100
6.1.6 (2x3x4) NERA Sub Array Indexing Analysis.....	102
6.2 Catenation definition and a Discussion of Lamination:	106
6.2.1 Catenation Shapeton Example (2x3x4) ,[3] (2x3x4).....	110
6.2.2 The Difference Between a (2x3x1) and a (2x3) Holding Equal Data.	112
6.2.3 Shapeton Meaning In Catenation.....	114
6.2.4 Lamination Definition [12; 5; 4] and Example.....	114
CHAPTER 7. APPLICATION OF THE EXPANSIONS.....	116
7.1 Array Division (/) as "Polynomial Division".	116
7.1.1 Examples	117
7.1.2 Division From the Other Direction	118
7.1.3 (2x3x4)\(1x2x4) and (2x3x4)/(1x2x4) Division Examples.	122
7.1.4 Shapetons as Remainders.....	124
7.1.5 Non-existence, in General of Multiplicative Array Inverses.....	129
7.2 Hyper cubes (HC) [23].....	130
7.2.1 Hyper cube Definitions and Examples.....	131
7.2.2 Generalized Hyper cubes (GHC) [3].....	138
7.2.3 Comment on HC and ragged arrays.....	141
7.2.4 A Short Dictionary of Array Decompositions.....	141
7.2.5 Two GHC (Ragged Array) Examples Using the ERA and NERA Expansions.	147
7.3 OLAP Data Hierarchies	158
7.3.1 Hierarchical Data Example (2x{2,1}x{2,2})	159

7.4 Applications in Computer Languages.....	164
7.4.1 Current Mathematics Usage of Null Data Structures.....	164
7.4.2 Current Computing Usage of Null Data Structures.....	164
7.4.3 Suggestion: New Array Indexing Rules for Computer Languages	178
CHAPTER 8. IMPLICATIONS FOR FURTHER RESEARCH.....	179
8.1 Array Valued Polynomials.....	179
8.1.1. (Independent Proof of the NERA Master Equation).....	179
8.1.2 The Full Binomial Series Theorem: (BS) [20].....	182
8.2 Complex Axis Length	183
8.2.1 Independent Proof of the ERA Expansion	183
8.2.2 Computing Lengths - a Non-Linear Problem.....	183
8.3 Fractional Lengths and the Binomial Series	184
8.4 Ragged Arrays.....	186
8.4.1 R3 Fractional Length Example.....	187
8.4.2 R4 Fractional Length Example.....	188
8.5 Complex and Fractional Dimensional Arrays.....	188
8.6 NERA – GHC Connectivity Machines.....	189
8.7 Unique Factorization Domains [22]	189
CHAPTER 9. SUMMARY	190
APPENDICES.....	191
A.1. Comments	191
A.1.1 Comment on the egg crate view	191
A.1.2 Comment on the Iverson Convention [$(0)^0 = 1$].....	191
A.1.3 Excel Spread Sheet Computing $(0)^0$	192
A.1.4 Adin Falkoff on the Word “Catenate”	194
A.1.5 Null Array Jokes.....	194
A.2 Coefficient Examples Computed by My Java Code	195
A.2.1 $T_0^0 = T_0^0$	196
A.2.2 $T_5^1 = 5T_0^0 + T_0^1$	196

A.2.4	$T_{3,3}^2 = 9T_0^0 + 6T_0^1 + T_0^2$	196
A.2.5	$T_{7,5,6}^3 = 210T_0^0 + 107T_0^1 + 18T_0^2 + T_0^3$	197
A.2.6	$T_{2,2,2,2}^4 = 16T_0^0 + 32T_0^1 + 24T_0^2 + 8T_0^3 + T_0^4$	198
A.2.7	$T_{2,3,4,1,5}^5 = 120T_0^0 + 274T_0^1 + 225T_0^2 + 85T_0^3 + 15T_0^4 + T_0^5$	199
A.3 Appendix III Java Code for Computing Coefficients of the Master Equation		202
BIBLIOGRAPHY		221

List of Figures

Figure 1. Logical Flow of Development	6
Figure 2. Logical Flow by Argument Type.....	7
Figure 3. Binomial Picture From Reference [9].....	10
Figure 4. Egg Crate.....	15
Figure 5. (2x2x2) Basic Array Informal Definitions.....	16
Figure 6. (2x2x2) Basic Array Formal Definitions.....	24
Figure 7. (2x2) First Level Decomposition.....	43
Figure 8. (3x3x3) First Level Decomposition - Indices Showing.	44
Figure 9. Preliminary Table of Null Array Results	52
Figure 10. (3x3x3) (n+1) Decomposition With Canceling Sub Arrays (A shell).	57
Figure 11. Triangular Summation (Horizontal).	60
Figure 12. Triangular Summation (Vertical).	61
Figure 13. (2x2x3) First Level NERA Decomposition.	72
Figure 14. (2x3x4) NERA Full Decompsition.	94
Figure 15. (2x2x2) ERA Solid View Full Decomposition.....	97
Figure 16. (2x2x2) ERA Wire Frame (Node View) Full Decomposition.....	98
Figure 17. (2x2x2) ERA Wire Frame Full Decomposition with Indices Showing. ..	99
Figure 18. (2x3x4) NERA Wire Frame Full Decomposition.	103
Figure 19. (4x4) Array Division Result.	120
Figure 20. (8x2) = (2x8) Array Division Result.	121
Figure 21. (2x3x4) // (1x2x4) Division with Shapetons.....	126

Figure 22. 0-D, 1-D, & 2-D Hyper cube Wire Frame View.....	131
Figure 23. 0-D, 1-D, 2-D, & 3-D Hyper cube Wire Frame View.	132
Figure 24. 3-D Hyper cube Wire Frame View.	133
Figure 25. 4-D Hyper cube Wire Frame View.	134
Figure 26. 5-D Hyper cube Wire Frame View.	135
Figure 27. 5-D Hyper cube Projected Tesseract View.....	136
Figure 28. The null (no cell) array. $0T_0^0 = 0$	141
Figure 29. The single node or cell. $1T_0^0 = T_0^0$	141
Figure 30. The 2-node or 2-cell. $2T_0^0$	142
Figure 31. The vector of length 0. $1T_0^1 = 0T_0^0 + T_0^1$	142
Figure 32. The vector of length 1. $T_1^1 = 1T_0^0 + T_0^1$	142
Figure 33. The vector of length 2. $T_2^1 = 2T_0^0 + T_0^1$	142
Figure 34. Two vectors of length 1. $2T_1^1 = 2T_0^0 + 2T_0^1$	143
Figure 35. The 2-D cell of length 1. $T_{1,1}^2 = 1T_0^0 + 2T_0^1 + T_0^2$	143
Figure 36. The 2-D array (1x2). $T_{1,2}^2 = 2T_0^0 + 3T_0^1 + T_0^2$	143
Figure 37. The 2-D array (2x2). $T_{2,2}^2 = 4T_0^0 + 4T_0^1 + T_0^2$	144
Figure 38. The 2-D array (2x3). $T_{2,3}^2 = 6T_0^0 + 5T_0^1 + T_0^2$	144
Figure 39. Catenation of Two Vectors of Length 1 $2T_0^1 + T_0^1$	145

Figure 40. The 3-D Cell (1x1x1)	$T_1^3 \triangleq T_{1,1,1}^3 = 1T_0^0 + 3T_0^1 + 3T_0^2 + T_0^3$	145
Figure 41. “Hyper cube” (2x2x2)	$T_2^3 \triangleq T_{2,2,2}^3 = 8T_0^0 + 12T_0^1 + 6T_0^2 + T_0^3$	146
Figure 42. 3-D (2x2x1)	$T_{2,2,1}^3 = 4T_0^0 + 8T_0^1 + 5T_0^2 + T_0^3$	147
Figure 43. (1x1x1) Wire form.		148
Figure 44. (2x2x2) - (1x1x1).		149
Figure 45. Intermediate step.		150
Figure 46. Penultimate step.		151
Figure 47. Two Isolated Nodes.		151
Figure 48. (2x3) 6 Step Summary.		152
Figure 49. (2x3) Step 1.		154
Figure 50. (2x3) Step 2.		155
Figure 51. (2x3) Step 3.		156
Figure 52. (2x3) Step 4.		156
Figure 53. (2x3) Step 5.		157
Figure 54. (2x3) Step 6.		157
Figure 55. Relationships: Sub cube Allocation, Ragged Arrays, ERA and NERA		
Expansions.		158
Figure 56. OLAP Data Aggregation Hierarchy (2x3x4)→(2x{2,1}x{2,2})		160
Figure 57. APL2 Output Screen Showing Null Array Catenation.		166
Figure 58. JavaDoc 1.3 for newInstance() constructor allowing 0-length arrays..		167
Figure 59. Java Code Constructing a 0-Length Array.		170

Figure 60. JavaDoc 1.3 for other data structures allowing null content (“empty”).	
.....	171
Figure 61. JavaDoc 1.3 Update to a Complete Set of Empty Data Structures.....	171
Figure 62. My C++ code showing C++ does not handle null arrays.	178
Figure 63. The Complete Binomial Series Theorem for Complex Base and	
Exponent.	182
Figure 64. (4x4x4) minus (3x3x3) ragged array “shell”.	185
Figure 65. Ragged arrays $R_4 + R_3 = (4x4)$.	187
Figure 66. (4x8) “Egg Crate”.	191
Figure 67. Excel graph of $(0)^0 = 1$.	194
Figure 68. $T_0^0 = T_0^0$ Java Code output.	196
Figure 69. $T_5^1 = 5T_0^0 + T_0^1$ Java Code output.	196
Figure 70. $T_{3,3}^2 = 9T_0^0 + 6T_0^1 + T_0^2$ Java Code output.	197
Figure 71. $T_{7,5,6}^3 = 210T_0^0 + 107T_0^1 + 18T_0^2 + T_0^3$ Java Code output.	198
Figure 72. $T_{2,2,2,2}^4 = 16T_0^0 + 32T_0^1 + 24T_0^2 + 8T_0^3 + T_0^4$ Java Code output.	199
Figure 73. $T_{2,3,4,1,5}^5 = 120T_0^0 + 274T_0^1 + 225T_0^2 + 85T_0^3 + 15T_0^4 + T_0^5$ Java Code output.	
.....	201
Figure 74. My Java Code main().	203
Figure 75. My Java Code GetCoeffs.	206
Figure 76. My Java Code NtoBinFunc ().	208

Figure 77. My Java Code RIFmath.....	210
Figure 78. My Java Code ListWriter.....	212
Figure 79. My Java Code ListElement.....	216
Figure 80. My Java Code ListComparator.....	217
Figure 81. Java Code by D. Liang MyInput – for Input of N and the n_i	220

PREVIEW

List of Tables

Table 1. Preliminary Table of Null Array Results.....	52
Table 2. Some Coefficients of $(n+1)^N$.....	66
Table 3. Final Table of Results of ERA First Level Decompositions	67
Table 4. Hyper cube Nodes, Edges, and Incidence Count.....	137

PREVIEW