

Emergence, Complexity and Computation

Volume 18

Series editors

Ivan Zelinka, Technical University of Ostrava, Ostrava, Czech Republic
e-mail: ivan.zelinka@vsb.cz

Andrew Adamatzky, University of the West of England, Bristol, UK
e-mail: adamatzky@gmail.com

Guanrong Chen, City University of Hong Kong, Hong Kong, China
e-mail: eegchen@cityu.edu.hk

Editorial Board

Ajith Abraham, MirLabs, USA

Ana Lucia C. Bazzan, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brazil

Juan C. Burguillo, University of Vigo, Spain

Sergej Čelikovský, Academy of Sciences of the Czech Republic, Czech Republic

Mohammed Chadli, University of Jules Verne, France

Emilio Corchado, University of Salamanca, Spain

Donald Davendra, Technical University of Ostrava, Czech Republic

Andrew Ilachinski, Center for Naval Analyses, USA

Jouni Lampinen, University of Vaasa, Finland

Martin Middendorf, University of Leipzig, Germany

Edward Ott, University of Maryland, USA

Linqiang Pan, Huazhong University of Science and Technology, Wuhan, China

Gheorghe Păun, Romanian Academy, Bucharest, Romania

Hendrik Richter, HTWK Leipzig University of Applied Sciences, Germany

Juan A. Rodriguez-Aguilar, IIIA-CSIC, Spain

Otto Rössler, Institute of Physical and Theoretical Chemistry, Tübingen, Germany

Vaclav Snasel, Technical University of Ostrava, Czech Republic

Ivo Vondrák, Technical University of Ostrava, Czech Republic

Hector Zenil, Karolinska Institute, Sweden

About this Series

The Emergence, Complexity and Computation (ECC) series publishes new developments, advancements and selected topics in the fields of complexity, computation and emergence. The series focuses on all aspects of reality-based computation approaches from an interdisciplinary point of view especially from applied sciences, biology, physics, or chemistry. It presents new ideas and interdisciplinary insight on the mutual intersection of subareas of computation, complexity and emergence and its impact and limits to any computing based on physical limits (thermodynamic and quantum limits, Bremermann's limit, Seth Lloyd limits...) as well as algorithmic limits (Gödel's proof and its impact on calculation, algorithmic complexity, the Chaitin's Omega number and Kolmogorov complexity, non-traditional calculations like Turing machine process and its consequences,...) and limitations arising in artificial intelligence field. The topics are (but not limited to) membrane computing, DNA computing, immune computing, quantum computing, swarm computing, analogic computing, chaos computing and computing on the edge of chaos, computational aspects of dynamics of complex systems (systems with self-organization, multiagent systems, cellular automata, artificial life,...), emergence of complex systems and its computational aspects, and agent based computation. The main aim of this series is to discuss the above mentioned topics from an interdisciplinary point of view and present new ideas coming from mutual intersection of classical as well as modern methods of computation. Within the scope of the series are monographs, lecture notes, selected contributions from specialized conferences and workshops, special contribution from international experts.

More information about this series at <http://www.springer.com/series/10624>

Paul Rendell

Turing Machine Universality of the Game of Life

 Springer

Paul Rendell
Department of Computer Science
University of the West of England
Bristol
UK

ISSN 2194-7287 ISSN 2194-7295 (electronic)
Emergence, Complexity and Computation
ISBN 978-3-319-19841-5 ISBN 978-3-319-19842-2 (eBook)
DOI 10.1007/978-3-319-19842-2

Library of Congress Control Number: 2015942788

Springer Cham Heidelberg New York Dordrecht London
© Springer International Publishing Switzerland 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

For Cheryl, Jack and Harry

Preface

I became interested in Conway's Game of Life in the 1980s when I wrote a Game of Life program for an Amstrad CPC 464. This little computer with 64K of memory ran on a Z80 microprocessor with a 4 MHz clock. The program had a closed universe of 64×128 cells. Just big enough to hold some complex patterns. It ran fast enough to make manual searches for interesting interactions practical. Over a number of years I found the fanout and takeout patterns described in this book. I developed the fanout pattern into the matrix addressable memory cell which just fitted into the 64×128 universe.

My interest in Conway's Game of Life was reawakened towards the end of the 1990s when I got an Internet connection and discovered the patterns that other people had found in the Game of Life. With these extra components and the very powerful program Life32 by Johan Bontes I realised that I had all the pieces to build a Turing machine based on the my memory cell. This I proceeded to do using two stacks for the Turing tape. The completed machine was published on the Internet in 2000. I claimed that the design of the Turing machine allowed it to be extended to make a universal Turing Machine.

Between 2009 and 2014 I rebuilt the Turing machine as a fully universal Turing machine. This involved increasing the size of the memory used for the finite state machine to hold the larger universal Turing Machine program and adding stack constructor patterns to build stack cells as the machine is running so that the Turing machine never runs out of blank tape.

In order to demonstrate the universal Turing machine it was important that the machine ran reasonably quickly. I therefore designed a universal Turing machine that runs in polynomial time by taking advantage of the available size of the finite state machine's memory. This universal Turing machine requires a description of the simulated Turing machine as a list of transitions that can be in any order.

While investigating the best ordering of this list I formulated the problem as a simple Quadratic Assignment Problem and found that I could solve this with a simple procedure. I was not able to prove formally that the Quadratic Assignment Problem had been solved but informally it is very convincing.

UWE Bristol
June 2014

Paul Rendell

Acknowledgments

My thanks to Prof. Andy Adamatzky for making this book possible and for help and guidance in overcoming the many obstacles along the way.

Many thanks to my colleagues at the University of the West of England: Larry Bull, Paul White and Genaro Martinez for their help with QAP work and with the Cellular Automata Workshop.

Thanks go to Suzanne Britton for providing me with a Turing machine simulator written in Java and allowing me to modify it.

Thanks go to Adam P. Goucher and Matthias Merzenich for finding the period 450 diagonal $c/5$ rakes for the diagonal stack constructor in the Game of Life.

I am indebted to many people who found and constructed the Game of Life patterns which they have made publicly available through the World Wide Web.

My thanks also go to my wife Cheryl and family, Jack and Harry, who have supported me and made sacrifices for me over this interesting and busy period of my life.

Contents

1	Introduction	1
	References.	3
2	Background	5
2.1	Conway’s Game of Life	5
2.2	Turing Machines.	9
2.2.1	Turing Machine Structure	9
2.2.2	Universal Turing Machines	10
2.2.3	Example Turing Machine.	11
2.3	Counter Machines.	14
2.4	Universality of the Game of Life	15
	References.	16
3	Literature Review/Related Work	17
3.1	Conway’s Winning Ways	17
3.2	Buckingham and Niemiec’s Adder	18
3.3	Dean Hickerson’s Sliding Block Memory	20
3.4	Paul Chapman’s Counter Machine	21
3.4.1	Machine Structure.	21
3.4.2	Counter Machine Operation	23
3.4.3	The First Few Instructions	24
3.4.4	The NOP Instruction	27
3.4.5	The Counter Module	27
3.4.6	The Universal Counter Machine	27
3.4.7	Statistics	29
3.5	Chapman’s P1 Universal Counter Machine	29
3.5.1	The Herschel and Herschel Tracks	33
3.5.2	Stable Reflector	33
3.5.3	Statistics	34

- 3.6 Spartan Universal Computer-Constructor 35
 - 3.6.1 Description 35
 - 3.6.2 Statistics 36
- 3.7 Rogozhin’s Universal Turing Machine. 36
 - 3.7.1 Universal 2-Tag System. 36
 - 3.7.2 Rogozhin’s 2-Tag UTM. 40
- 3.8 Weakly Universal Turing Machines. 41
 - 3.8.1 Neary and Woods 42
 - 3.8.2 Wolfram 43
- References. 43

- 4 Game of Life Turing Machine. 45**
 - 4.1 Construction of the Turing Machine 45
 - 4.2 Basic Patterns. 47
 - 4.2.1 Period Thirty: Queen Bee Shuttle Based 47
 - 4.2.2 Period Fifteen: Pentadecathlon Based 49
 - 4.3 The Finite State Machine. 52
 - 4.3.1 The Machine in the Pattern 54
 - 4.3.2 Memory Cell 54
 - 4.3.3 Address Comparator 54
 - 4.3.4 Selection of a Row 58
 - 4.3.5 Selection of a Column. 59
 - 4.3.6 Collecting Data from the Memory Cell 59
 - 4.4 Signal Detector/Decoder 60
 - 4.5 Stack. 63
 - 4.5.1 Stack Cells. 63
 - 4.5.2 Stack Control 64
 - 4.5.3 Conclusion. 70
 - References. 70

- 5 Game of Life Universal Turing Machine 71**
 - 5.1 Simple Universal Turing Machine (SUTM) 71
 - 5.1.1 SUTM Description 71
 - 5.1.2 STUM Results 75
 - 5.1.3 Running Time 75
 - 5.2 Expanding the Size of the Turing Machine 77
 - 5.2.1 Expanding the Stack 78
 - 5.2.2 Expanding the Finite State Machine 79
 - 5.2.3 Loading the Stack. 81
 - 5.2.4 Statisits 82
 - 5.3 Larger Example TM: Unary Multiplication. 83
 - 5.3.1 The Unary Multiplication TM 83
 - 5.3.2 Coding the Unary Multiplication TM 84
 - 5.4 Conclusion. 87
 - References. 89

- 6 Optimizing Transition Order.** 91
 - 6.1 Problem Definition 91
 - 6.2 SUTM Optimization Data 92
 - 6.3 Solution Method. 95
 - 6.4 Initial Analysis of Results 96
 - 6.5 Comparison with the Greedy Method 97
 - 6.6 Expected Basin of Attraction Size. 98
 - 6.6.1 Simulation for Expected Size 98
 - 6.6.2 Simulation for Expected Number of Neighbours. 99
 - 6.6.3 Time to Discovery 100
 - 6.7 Further Quantitative Analysis 101
 - 6.7.1 Splitting the Sample 101
 - 6.7.2 Analysis of Hard to Find Local Minima 101
 - 6.8 Conclusion. 103
 - 6.8.1 The SUTM 103
 - 6.8.2 The QAP Solution 104
 - References. 105

- 7 Forty Five Degree Stack** 107
 - 7.1 Dual Cell Design 108
 - 7.2 Main Cell Timing 109
 - 7.2.1 Pop Operation 109
 - 7.2.2 Push Operation. 110
 - 7.3 Delay Cell Timing 110
 - 7.4 Control Signal Generation 110
 - 7.5 Next State Delay 112
 - 7.6 Push/Pop Switch. 112
 - 7.7 Loading the Stack. 114
 - 7.8 Conclusion. 117
 - Reference 117

- 8 Stack Constructor.** 119
 - 8.1 Design Procedure 119
 - 8.2 The Construction Tool. 121
 - 8.2.1 Pattern Generation 123
 - 8.2.2 Input to the Construction Tool 126
 - 8.2.3 Phase III: Connecting a New Cell to the Stack 126
 - 8.2.4 Phase II: Activation. 126
 - 8.2.5 Phase I: Building 128
 - 8.3 Rake Convoys 129
 - 8.4 Alternative Rakes 133
 - 8.4.1 Diagonal C/12 Rakes 133
 - 8.4.2 Diagonal C/5 Rakes 137
 - 8.5 Conclusion. 141
 - References. 141

9 Universal Counter Machine—Turing Machine 143

9.1 Counter Machine Turing Machine Program 143

9.2 The Example Turing Machine 144

9.3 Statistics 145

References. 146

10 Wolfram’s Two State Three Symbol UTM. 147

References. 149

11 Conclusions and Discussion 151

11.1 The Turing Machine in the Game of Life 151

11.2 Demonstrating Universality 154

11.2.1 Universal Counter Machine in the Game of Life. 154

11.2.2 Universal Turing Machine in the Game of Life. 155

11.3 Quadratic Assignment Problems with Large Basins
of Attraction. 157

11.4 Formatting Blank Media 158

References. 158

12 Further Work. 161

12.1 Large Basins of Attraction in the QAP 161

12.2 Finite Turing Machines and Complexity 161

12.3 Construction Order Efficiency 162

References. 162

Appendix A: UCM: Counter Machine Program 163

Appendix B: UCM: Turing Machine Program 167

Appendix C: Tag Productions 171

Index 175

Acronyms

Chap.	Chapter
Fig.	Figure
FSM	Finite State Machine
Gb	Gigabyte
Gen.	Generation
GHz	Gigahertz
GoL	Game of Life
HWSS	Heavy Weight SpaceShip
LWSS	Light Weight SpaceShip
MWSS	Medium Weight SpaceShip
NP	Nondeterministic Polynomial
QAP	Quadratic Assignment Problem
RAM	Random Access Memory
Sect.	Section
SUTM	Simple Universal Turing Machine
TM	Turing Machine
UCM	Universal Counter Machine
UTM	Universal Turing Machine
UTS	Universal Tag System
XNOR	The Logic operation: the inverse of exclusive OR (XOR)
XOR	The Logic operation: exclusive OR