“MP Grammars, Reactive Systems and Electric Circuits”
(2nd year presentation)

Ricardo Henrique Gracini Guiraldelli
ricardo.guiraldelli@univr.it
http://ricardo.guiraldelli.com/

Università degli Studi di Verona

October 29th, 2014

Short Review

Metabolic P Systems

- it is a computational model inside Membrane Computing
- it is deterministic
- it is a dynamical system
- it is designed for real-world application (in Biology)
- it is simple (to read, to understand, to express process)
- ...

Metabolic P Systems

- it is a computational model inside Membrane Computing
- it is deterministic
- it is a dynamical system
- it is designed for real-world application (in Biology)
- it is simple (to read, to understand, to express process)
- ...

Metabolic P Systems

- it is a computational model inside Membrane Computing
- it is deterministic
- it is a dynamical system
- it is designed for real-world application (in Biology)
- it is simple (to read, to understand, to express process)
- ...

Metabolic P Systems: Simplicity

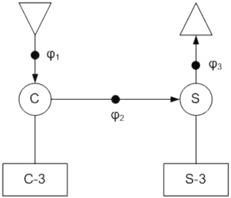
1. set of variables
2. set of rules (of how the variables interchanges)
3. set of ~~functions~~ fluxes (rate of variation of the rules)

Metabolic P Systems: Flavors

1. MP Grammars

$$
\begin{array}{lll}
r_1: & \emptyset \rightarrow C & \varphi_1 = 0.0030015 + 0.0009995 \times C \\
r_2: & C \rightarrow S & \varphi_2 = 0.001 \times C + 0.001 \times S \\
r_3: & S \rightarrow \emptyset & \varphi_3 = 0.0029985 + 0.0010005 \times S
\end{array}
$$

2. MP Graphs



[6]

Electric Circuits

- (mathematical) model of physical ones
- simplification of analysis (*e.g.* Maxwell's laws)
- simpler description: equations, block diagrams, schematic diagrams, ...
- transform signals into signals (reactive systems)

Electric Circuits

- (mathematical) model of physical ones
- simplification of analysis (*e.g.* Maxwell's laws)
- simpler description: equations, block diagrams, schematic diagrams, . . .
- transform signals into signals (reactive systems)

Electric Circuits: Flavors

1. analog circuits
   1.1 linear circuits (resistors, capacitors and inductors)
   1.2 nonlinear circuits (diodes, transistors, ...)
2. digital circuits
   2.1 combinational circuits (pure functions: $f(x) = y$, always)
   2.2 sequential circuits (memory units are *essential*[1])

[1] Does it remind you of Chomsky hierarchy? 😉

Electric Circuirs: Computational Power

- analog circuits: process real ($\mathbb{R}$) values, *continuously* (in time)
- digital circuits: fixed-precison ($\mathbb{Q}$) values, discretely (in time)

Electric Circuirs: Computational Power

- analog circuits: process real ($\mathbb{R}$) values, *continuously* (in time)
- digital circuits: fixed-precison ($\mathbb{Q}$) values, discretely (in time)

is that a set theoretical discussion?

$\aleph_1$ and infinite cardinality *versus* $\aleph_0$ and finite cardinality?

Electric Circuirs: Computational Power

- analog circuits: process real ($\mathbb{R}$) values, *continuously* (in time)
- digital circuits: fixed-precison ($\mathbb{Q}$) values, discretely (in time)

is that a set theoretical discussion?

$\aleph_1$ and infinite cardinality *versus* $\aleph_0$ finite representation and finite cardinality?

Research Proposal: 2013

$$\xi(\mathit{Cell}) = \mathit{Circuit}$$

$$\exists \xi, \zeta ?\ \zeta = \xi^{-1} ?$$

$$\zeta(\mathit{Circuit}) = \mathit{Cell}$$

Research Proposal: 2013

demonstration of a mathematical equivalence between dynamics described by Metabolic P systems and electronics circuits

The Activities (Since Then)

Disclamer

- activities may look sparse, but they have (or had) a logical track
- they are presented in chronological order in order to understand choices
- details are ommitted (for everybody's sake) without compromising the argument
- it is not expected that *anyone* master all the fields involved $\Longrightarrow$ questions shouldn't be taken further

Disclamer

- activities may look sparse, but they have (or had) a logical track
- they are presented in chronological order in order to understand choices
- details are ommitted (for everybody's sake) without compromising the argument
- it is not expected that *anyone* master all the fields involved $\Longrightarrow$ questions shouldn't be taken further
- if something looks too obvious (meaning *"easy"*), probably some important argument was lost or misunderstand (or am I a great presenter?)

Disclamer

- activities may look sparse, but they have (or had) a logical track
- they are presented in chronological order in order to understand choices
- details are ommitted (for everybody's sake) without compromising the argument
- it is not expected that *anyone* master all the fields involved $\implies$ questions shouldn't be taken further
- if something looks too obvious (meaning *"easy"*), probably some important argument was lost or misunderstand (or am I a great presenter?)
  - by the other side, things *shouldn't* be too complicated

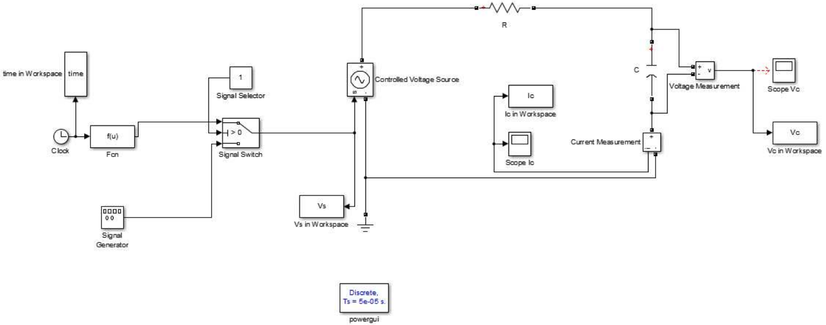
Analysis on Analog Circuits: Why?

- the real question is: *why not?*
- models integro-differential equations
- highest computational power [10]
- fast performance [12]
- agenda of diverse researchers [3, 2, 8, 15]
- MP has (sine and cosine) oscillators as well as analog circuits

Analysis on Analog Circuits: How? (1)

- reverse engineering
- simple analog circuits: oscillators such as RC, RL, RLC, . . .
- Log Gain Stoichiometric Stepwise (LGSS) regression algorithm

Analysis on Analog Circuits: How? (2)



Analysis on Analog Circuits: How? (3)

Name of the Dictionary	**Composition Rule**
All Functions	$D = F$
Trigonometric	$D = \{1, x, \sin(x), \cos(x)\}$
Complex Functions	$D = \{1, x, \exp(x), \frac{1}{1+\exp(-x)}, \lfloor x \rfloor\}$
Polynomial Functions	$D = \{1, x^n\}$ and redefined $n \in \{1, 2, 3, 4, 5\}$
Polynomial and Trigonometric Functions	$D = \{1, x^n, \sin(x), \cos(x)\}$
Composition of All Functions	$D = F \cup C$
Composition of Complex Functions	$D = F \cup C$ and redefined $F = \{1, x, \exp(x), \frac{1}{1+\exp(-x)}, \lfloor x \rfloor\}$
Composition of Trigonometric Functions	$D = F \cup C$ and redefined $F = \{1, x, \sin(x), \cos(x)\}$
Complete (All Possible Combinations)	$D = F \cup C \cup I$

where

$$\begin{cases} n \in \{1, 2, 3\} \\ x \in \{V_s, V_c, I_c\} \\ F = \{1, x^n, \sin(x), \cos(x), \exp(x), \frac{1}{1+\exp(-x)}, \lfloor x \rfloor\} \\ C = \{f \circ g : f, g \subseteq F\} \\ I = \{\frac{1}{h} : h \in F \cup C \setminus x\} \end{cases}$$

Analysis on Analog Circuits: What?

- few correct inferences (basically on sine/cosine)
- unable to deal with transient
- complicated and complex for input signals different from sine or cosine (still, complex fluxes)
- no silver bullet: non-linear optimization, theory-based dictionaries, fast Fourier transform analysis

Analysis on Analog Circuits: What?

- few correct inferences (basically on sine/cosine)
- unable to deal with transient
- complicated and complex for input signals different from sine or cosine (still, complex fluxes)
- no silver bullet: non-linear optimization (CMA-ES algorithm), theory-based dictionaries (Kolmogorov-Gabor polynomials or Maclaurin series), FFT analysis

Analysis on Analog Circuits: What?

- few correct inferences (basically on sine/cosine)
- unable to deal with transient
- complicated and complex for input signals different from sine or cosine (still, complex fluxes)
- no silver bullet: non-linear optimization, theory-based dictionaries, FFT analysis
- **good idea:** MP produces sine and cosine which are the basis of harmonic analysis, could MP do FFT of the signal and then infer the circuit?

Analysis on Analog Circuits: What?

- few correct inferences (basically on sine/cosine)
- unable to deal with transient
- complicated and complex for input signals different from sine or cosine (still, complex fluxes)
- no silver bullet: non-linear optimization, theory-based dictionaries, FFT analysis
- **good idea:** MP produces sine and cosine which are the basis of harmonic analysis, could MP do FFT of the signal ~~and then infer the circuit~~?

MP version of FFT: Why?

- because our approach to analog circuits failed
- but electrical engineers use FFT and Laplace transform to analyze these circuits
- because we know all (periodic) signals may be decomposed in harmonic (Fourier) series
- MP is able to produce sine and cosine with a simple grammar (and we made it generates other frequencies)
- regression with MP-I/O is good to point if a variable (rule) is need in the system

MP version of FFT: Why?

- because our approach to analog circuits failed
- but electrical engineers use FFT and Laplace transform to analyze these circuits
- because we know all (periodic) signals may be decomposed in harmonic (Fourier) series
- MP is able to produce sine and cosine with a simple grammar (and we made it generates other frequencies)
- regression with MP-I/O is good to point if a variable (rule) is need in the system

everything seems to fit together in this approach

MP version of FFT: How? (1)

1. verified that MP generates sines and cosines in different frequencies
2. generate (big) range of sines and cosines (memoization as time series)
3. regression algorithm in the input signal using the harmonic curves of different frequencies as variables

MP version of FFT: How? (1)

1. verified that MP generates sines and cosines in different frequencies
2. generate (big) range of sines and cosines (memoization as time series)
3. regression algorithm in the input signal using the harmonic curves of different frequencies as variables

⚠ ensure your results are not “addicted” to your solution

MP version of FFT: How? (1)

1. verified that MP generates sines and cosines in different frequencies
2. generate (big) range of sines and cosines (memoization as time series)
3. regression algorithm in the input signal using the harmonic curves of different frequencies as variables
4. implemented diverse versions of the algorithm

MP version of FFT: How? (2)

- generates sines and cosines via MP
- generates sines and cosines via MATLAB
- has a fixed-range of frequencies
- computes the range of frequencies dynamically (τ of MP and Nyquist frequency)

MP version of FFT: What? (1)

- success
- accuracy benchmark is better than MATLAB/FFTW
- speed benchmark isn't so promising, one order of magnitude slower
    - MP is "heavier" than *divide-and-conquer* strategy of FFT
    - interpreted code (MATLAB) and virtual machine (JVM) underperforms when compared to function (MATLAB's FFT) backed by native code (FFTW)
- project suspended for period abroad 

MP version of FFT: What? (2)

Signals	Frequency	Numerical Frequency	FFT	MP-FFT	MP-FFT (MATLAB)
1	20	20	20	20	20
2	$20 + df$	20.5	20.5	20.5	20.5
3	$20 + \frac{3}{4} \cdot df$	20.375	20.5	{20, 20.5}	{20 20.5}
4	$\{20, 47\}$	$\{20, 47\}$	{20, 47.5}	{20, 47, 47.5}	{20 47}
5	$\{20 + df, 47 + 3 \cdot df\}$	$\{20.5, 48.5\}$	{20.5, 49}	{20.5, 48.5, 49}	{20.5 48.5}
6	$\{20 + \frac{3}{4} \cdot df, 47 + \frac{2}{3} \cdot df\}$	$\{20.375, 47.\overline{3}\}$	{20.5, 47.5}	{20.5, 47.5}	{20.5 47.5}
7	20 + noise	20 + noise	{2, 3, 5, 7, 8.5, 10, 12, 14, 15, 16.5, 17.5, 18.5, 20, 22.5, 23.5, 24.5, 25.5, 26.5, 27.5, 29, 30.5, 32.5, 34, 35, 36.5, 38, 39.5, 41.5, 44, 45, 47, 48}	20	20
8	$20 + df$ + noise	20.5 + noise	{1, 3.5, 4.5, 6, 7.5, 10, 12, 14, 15.5, 17.5, 19, 20.5, 21.5, 23, 24, 25.5, 27, 28, 30, 32, 33, 34.5, 36, 37.5, 40, 42, 43, 44, 47, 48, 49.5}	20.5	20.5
9	$20 + \frac{3}{4} \cdot df$ + noise	20.375 + noise	{1.5, 2.5, 5.5, 6.5, 8, 9.5, 11, 12, 13, 15, 16, 17.5, 20.5, 22.5, 23.5, 25.5, 26.5, 27.5, 28.5, 29.5, 31, 32, 33.5, 34.5, 36, 37, 38, 40, 42, 43, 44.5, 46, 47, 49}	20.5	20.5

Digital Circuits: Why?

- because it is a specialization of electric circuits (should be simpler)
- because there is an equivalence hardware-software [14, 13]
- because it is a discrete system such as MP
- because the research proposal has origin in some ideas from digital circuits
- because its component analysis is easier
- because its design is easier (great quantity of tools)
- *because the group I was worked with it*

VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS

Digital Circuits: How? (1)

Implementation (or *The Short Road*)

- implemented MP systems in VHDL
- converted VHDL to FPGA
- “REPL”: read, evaluate, program, loop
- model systems by components (arithmetical)

Digital Circuits: How? (1)

Implementation (or *The Short Road*)

- implemented MP systems in VHDL
- converted VHDL to FPGA
- "REPL": read, evaluate, program, loop
- model systems by components (arithmetical)
  - before: attempts using previous experience
  - later: arithmetical network

Digital Circuits: How? (2)

Formal Proof (or *The Long Road*)

- pencil and paper
- trial and error
- spanned through fields such as: dynamical systems, automata theory, computability, logic synthesis, control theory, algebraic groups, commutative diagrams, category theory, to name a few
- communication with other researchers (which gave me different perspectives and a solution)

Digital Circuits: How? (2)

Formal Proof (or *The Long Road*)

- pencil and paper
- trial and error
- spanned through fields such as: dynamical systems (influenced by Hinrichsen and Pritchard [4]), automata theory, computability, logic synthesis, control theory, algebraic groups, commutative diagrams, category theory, to name a few
- communication with other researchers (which gave me different perspectives and a solution)

Digital Circuits: What? (1)

Implementation (or *The Short Road*)

- success [2]
- working hardware that performs MP dynamics (including its accumulated errors!)
- framework for component-level design of MP systems (arithmetical network)

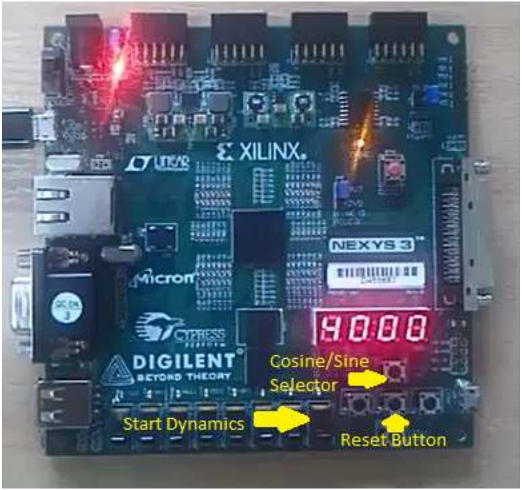
Digital Circuits: What? (1)

Implementation (or *The Short Road*)

- success [2]
- working hardware that performs MP dynamics (including its accumulated errors!)
- framework for component-level design of MP systems (arithmetical network)

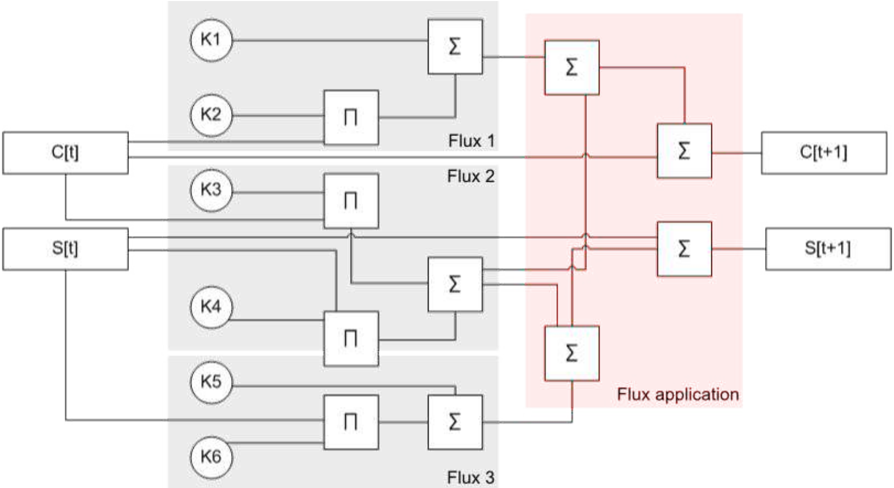
Digital Circuits: What? (1)

Implementation (or *The Short Road*)



Digital Circuits: What? (1)

Implementation (or *The Short Road*)



Digital Circuits: What? (2)

Formal Proof (or *The Long Road*)

- two *almost* proofs: diagram chasing and constructive one
- representation of both MP and digital circuits as dynamical systems [4, Definition 2.1.1]

Digital Circuits: What? (2)

Formal Proof (or *The Long Road*)

- two *almost* proofs: diagram chasing and constructive one
- representation of both MP and digital circuits as dynamical systems [4, Definition 2.1.1]

$$
\begin{array}{ccccccc}
T_{\mathcal{M}} & \qquad & U_{\mathcal{M}} & \xrightarrow{\Phi} & X_{\mathcal{M}} & \xrightarrow{id} & Y_{\mathcal{M}} \\
\downarrow{\scriptstyle id} & & \downarrow{\scriptstyle round} & & \downarrow{\scriptstyle \mathbf{?}} & & \downarrow{\scriptstyle round} \\
T_{\mathcal{D}} & & U_{\mathcal{D}} & \xrightarrow{\Phi_{\mathcal{D}}} & X_{\mathcal{D}} & \xrightarrow{id} & Y_{\mathcal{D}}
\end{array}
$$

Digital Circuits: What? (2)

Formal Proof (or *The Long Road*)

- two *almost* proofs: diagram chasing and constructive one
- representation of both MP and digital circuits as dynamical systems [4, Definition 2.1.1]
- questions about MP dynamics
  - convergence of the systems
  - generalization of the definition of MP
  - *carriers* of accumulative errors

Digital Circuits: What? (2)

Formal Proof (or *The Long Road*)

- two *almost* proofs: diagram chasing and constructive one
- representation of both MP and digital circuits as dynamical systems [4, Definition 2.1.1]
- questions about MP dynamics
  - convergence of the systems
  - generalization of the definition of MP
  - *carriers* of accumulative errors

all of this before my return

Digital Circuits: What? (3)

Formal Proof (or *The Long Road* after return)

- change of perspective: computational equivalence
- return to previous works of the group [7]
- MP $\Leftrightarrow$ Register Machine $\wedge$ digital circuits $\Leftrightarrow$ UTM $\implies$ MP $\Leftrightarrow$ digital circuits[2]

[2] By the Church-Turing thesis [5, 11, § 5.1; p. 181]. 🙂

Digital Circuits: What? (3)

Formal Proof (or *The Long Road* after return)

- change of perspective: computational equivalence
- return to previous works of the group [7]
- MP $\Leftrightarrow$ Register Machine $\wedge$ digital circuits $\Leftrightarrow$ UTM $\implies$ MP $\Leftrightarrow$ digital circuits[2]

still ongoing work, nonetheless!

[2] By the Church-Turing thesis [5, 11, § 5.1; p. 181]. 🙂

Digital Circuits: What? (3)

Formal Proof (or *The Long Road* after return)

- change of perspective: computational equivalence
- return to previous works of the group [7]
- MP $\Leftrightarrow$ Register Machine $\wedge$ digital circuits $\Leftrightarrow$ UTM $\implies$ MP $\Leftrightarrow$ digital circuits[2]

still ongoing work, nonetheless!
(but already a good result! 😀)

[2] By the Church-Turing thesis [5, 11, § 5.1; p. 181]. 🙂

Personal Perspectives on the Research

Personal Perspectives on the Research

- from solely mathematics to mathematics, computer science, electrical engineering
- initial scope practically reached, but we're pushing for more (algorithms or methodologies for automatic translation)
- no wasted time: looking back, knowledge and results are impressive (lacking to publish)
- keep the wheel turning, finish projects on suspension
- but focus on the core of the research
- real-world is asking for similar solutions and I venture to say we're pavimenting the way of a new research field (impressions from SSBSS 2014 and state-of-the-art researches [8, 9, 1])
- pursuit of theoretical results is risky, but it is incredibly rewarding

Research Proposal: 2014

demonstration of a mathematical equivalence between dynamics described by Metabolic P systems and electronics circuits, *particularly digital ones*

The End

Thank you!

Grazie!

Ačiū!

Obrigado!

References (1)

Jacob Beal, Ting Lu, and Ron Weiss.
Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks.
PloS one, 6(8):e22490, January 2011.

Cristian Sorin Calude and Gheorghe Pun.
Bio-steps beyond Turing.
Bio Systems, 77(1-3):175–94, November 2004.

Abraham Gutiérrez, Luís Fernández, Fernando Arroyo, and Santiago Alonso.
Hardware and Software Architecture for Implementing Membrane Systems: A Case of Study to Transition P Systems.
In MaxH. Garzon and Hao Yan, editors, *DNA Computing SE - 22*, volume 4848 of *Lecture Notes in Computer Science*, pages 211–220. Springer Berlin Heidelberg, 2008.

Diederich Hinrichsen and Anthony J. Pritchard.
Mathematical Systems Theory I: Modelling, State Space Analysis, Stability and Robustness, volume 48 of *Texts in Applied Mathematics*.
Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

Harry Lewis and Christos Papadimitriou.
Elements of the Theory of Computation.
Prentice-Hall, Upper Saddle River, 2nd ed. edition, 1997.

References (2)

Vincenzo Manca.
Infobiotics: Information in Biotic Systems, volume 3 of *Emergence, Complexity and Computation*.
Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

Vincenzo Manca and Rosario Lombardo.
Computing with Multi-membranes.
In Marian Gheorghe, Gheorghe Pun, Grzegorz Rozenberg, Arto Salomaa, and Sergey Verlan, editors, *Membrane Computing*, volume 7184 of *Lecture Notes in Computer Science*, pages 282–299. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

R Sarpeshkar.
Analog synthetic biology.
Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 372(2012):20130110, March 2014.

Rahul Sarpeshkar.
Ultra-Low Power Bioelectronics. 1.
Cambridge University Press, 1st ed. edition, 2010.

Hava T. Siegelmann.
Analog Computational Power.
Science, 271(5247):373–373, January 1996.

Michael Sipser.
Introduction to the Theory of Computation.
Cengage Learning, Boston, USA, 3rd ed. edition, 2012.

References (3)

Elena Tamaševičit, Gytis Mykolaitis, and Arnas Tamaševičius.
Analog Modeling Complex Dynamical Systems Using Simple Electrical Circuits.
Proceedings of the 5th International Multi-Conference on Society, Cybernetics and Informatics, 3:236—-241, 2011.

Bernard Tiong Gie Tan.
Hardware and software equivalence.
International Journal of Electronics, 47(6):621–622, December 1979.

Kuo-Pao Yang and Theresa Beaubouef.
Equivalence of Hardware and Software: A case study for solving polynomial functions.
In *2010 42nd Southeastern Symposium on System Theory (SSST 2010)*, pages 318–322. IEEE, March 2010.

A. Steven Younger, Emmett Redd, and Hava T. Siegelmann.
Development of Physical Super-Turing Analog Hardware.
In Oscar H. Ibarra, Lila Kari, and Steffen Kopecki, editors, *Unconventional Computation and Natural Computation*, volume 8553 of *Lecture Notes in Computer Science*, pages 379–391. Springer International Publishing, London, ON, Canada, 2014.