Emulation networks of minimalistic universes

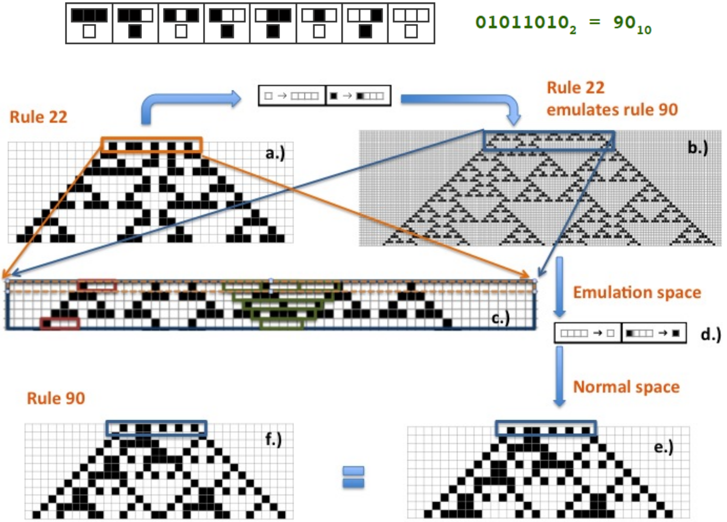
Jürgen Riedel[1]
Institut für Physik, Universität Oldenburg, Germany & Algorithmic Nature Group, LABORES, Paris
Hector Zenil[2]
Department of Computer Science, University of Oxford, UK & Karolinska Institute, Sweden & Algorithmic Nature Group, LABORES, Paris

http://arxiv.org/abs/1510.01671

Groningen, October 7th, 2015

[1]jurgen.riedel@labores.eu
[2]hector.zenil@algorithmicnaturelab.org

Reprogramming a CA rule through initial condition



Wolfram's Behavioural Classes (of Cellular Automata)

1. **Class I**: CA evolving to a **homogeneous state,** i.e. dominated by a unique state of its alphabet for any random initial condition.
2. **Class II**: CA evolving **periodically**, i.e. dominated by blocks of cells which are periodically repeated for any random initial condition.
3. **Class III**: CA evolving **chaotically**, i.e. for a long time and for any random initial condition, the evolution is dominated by sets of cells without any defined pattern.
4. **Class IV**: Includes all previous cases, known as a **class of complex rules**, i.e. the evolution is dominated by non-trivial structures emerging and travelling along the evolution space where uniform, periodic, or chaotic regions can coexist with these structures.

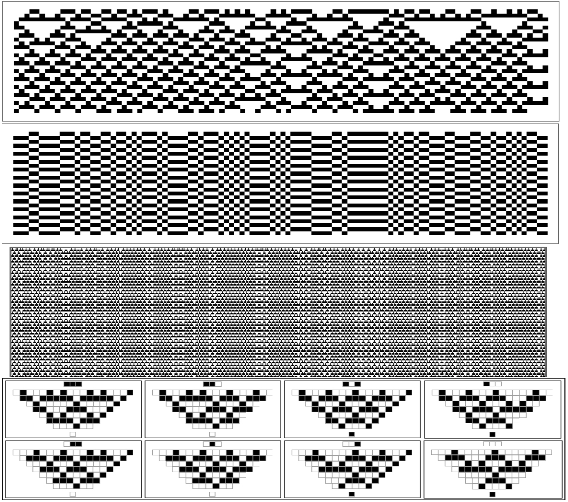
Reprogramming a class 4 cellular automaton (block size 6)

rule 54

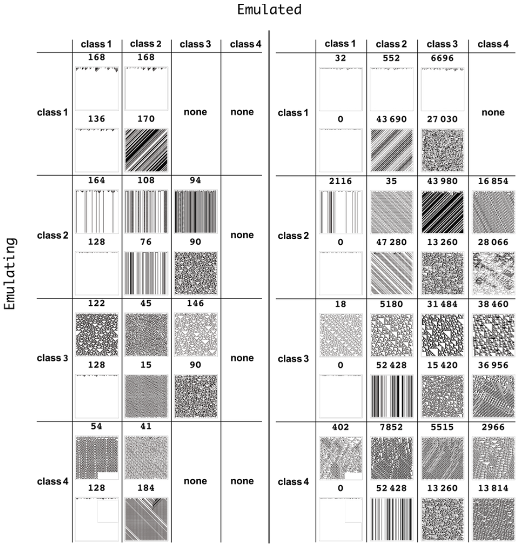
rule 51 (coarse grained)

emulation of rule 51

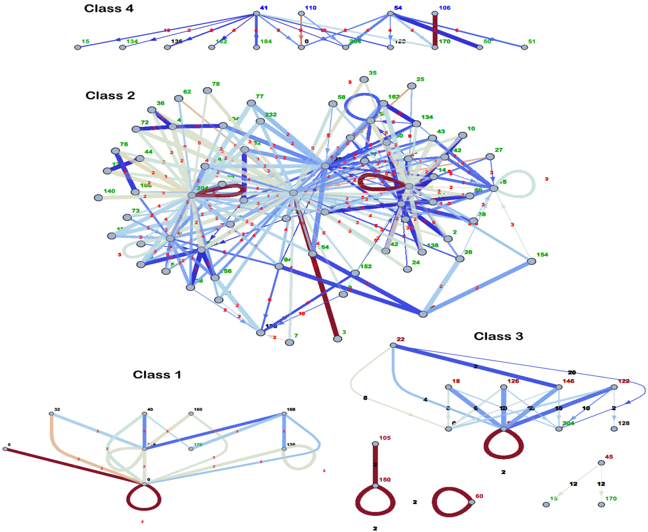
compiler



Cross-boundary ECA and GCA rules

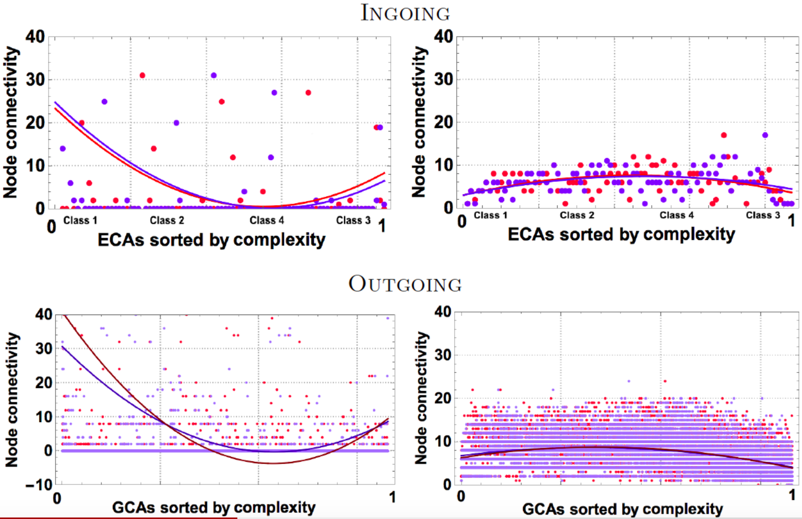


Rule emulation graph of the ECA rule space

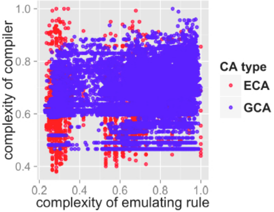


Topological Complexity of ECAs and GCAs

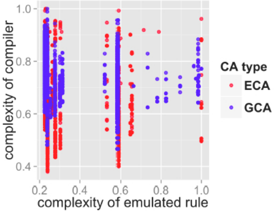
Outgoing edge degree | Ingoing edge degree



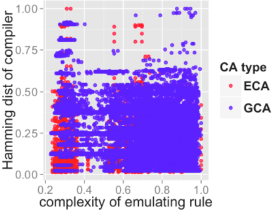
Exploring the complexity of the compiler space



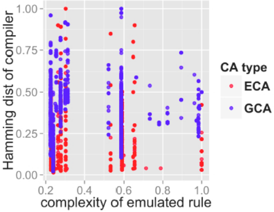
(a)



(b)



(c)



(d)

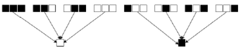
Rule composition

1. We have looked at **reprogramming initial conditions** of CAs using a compiler.
2. Now we look at the rules themselves and consider a **rule composition for CA tuples**.
3. $\text{rule C} = \text{rule A} \circ \text{rule B}$, i.e. the **lattice output of rule A is the input of rule B**.
4. We look **interaction of rules** to examine emulation capacities of CA tuples.
5. We ask which is the **minimal ECA rule set** which produces all other ECA rules by composition.

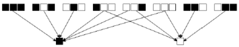
List of ECA 'prime' rules up to tuple size 3

Class 1	Class 2	Class 3
32	30	3, 9, 10, 14, 15, 19
40	60	23, 25, 26, 27, 28, 29
104	90	33, 35, 36, 37, 38, 43
160	105	33, 35, 36, 37, 38, 43
168		57, 58, 62, 73, 76, 77
		94, 108, 134, 154, 170

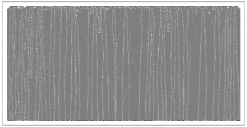
Table 1: List of prime ECA rules and their Wolfram class.



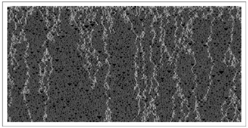
(a) Rule mapping ECA rule 54.



(c) Rule mapping rule 50 ∘ 37.

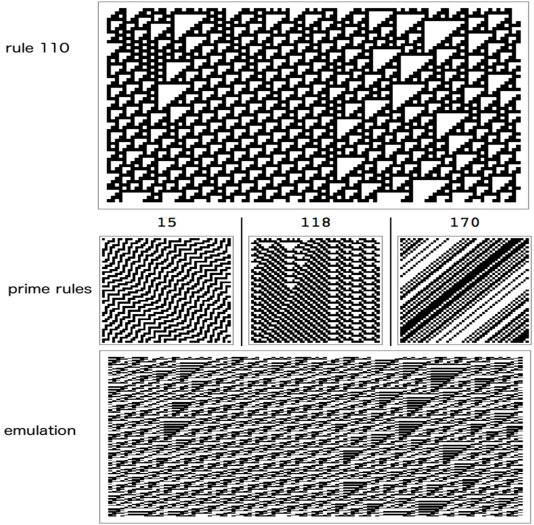


(b) Time evolution of ECA rule 54.



(d) Time evolution of rule 50 ∘ 37.

Emulation of rule 110



Wrap up

http://arxiv.org/abs/1510.01671

1. Introducing a **novel framework** for investigation at the **intersection of graph theory, complexity and cellular automata**.
2. Perhaps even the **simplest rules** may be capable of high complexity and possibly even **Turing universal computation** under Wolfram's own arguments and his **Principle of Computational Equivalence**
3. Opening of a **formal direction** the challenge of quantifying what seems to be a phenomenon of **ubiquitous computation universality**.
4. If **computer programs** such as cellular automata are taken as **toy models of digital universes**, the results our work strongly suggest that the **initial conditions are much more important than the rules**.
5. Even the **simplest physical laws** seem to be able to **emulate much more sophisticated universes** given the **proper initial condition**

Thank You!