Classification and Complexity

Automata 2007

Klaus Sutner
Carnegie Mellon University
www.cs.cmu.edu/~sutner

Classification

The Classification Problem

Come up with a taxonomy of CA that organizes them in a coherent, comprehensive way.

- dimension
- finiteness
- classes of configurations
- grid topology

Definitional level, underlying configuration space; not really part of the classification.

The Global Map

Fix configuration space and consider the global map.

- reversibility
- degree of irreversibility
- openness
- surjectivity
- local structures in diagram

Require only boundedly many applications of the global map.

Wolfram's Classes

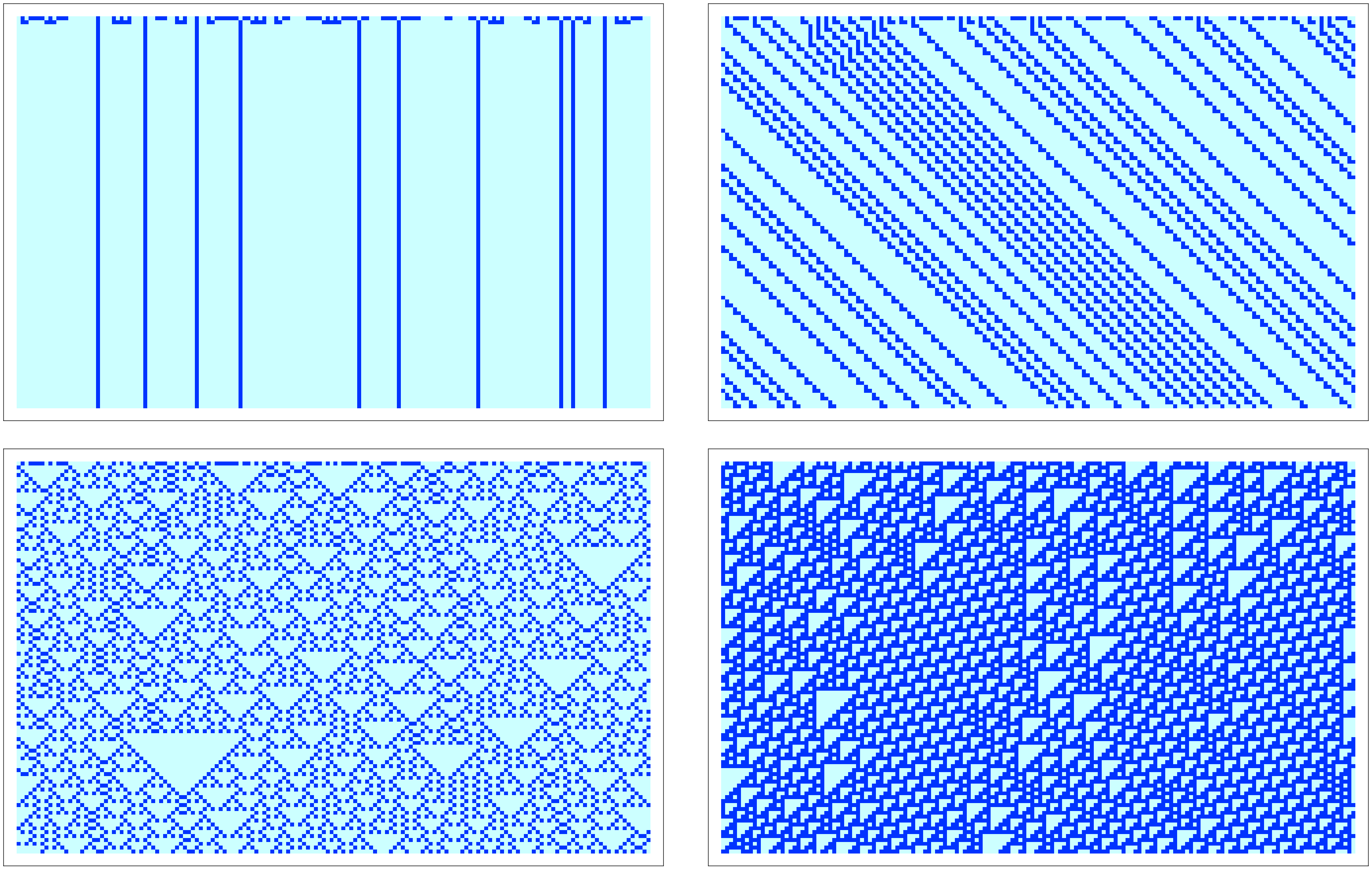
Aka iterating the global map.

- $W1$: Evolution leads to homogeneous fixed points.
- $W2$: Evolution leads to periodic configurations.
- $W3$: Evolution leads to chaotic, aperiodic patterns.
- $W4$: Evolution produces persistent, complex patterns of localized structures.

Require unboundedly many applications of the global map.

Appeal to visual characteristics of the orbits.

In Pictures



Automatic Classification

. . . is very hard, here are a few good news items.

- Amoroso and Patt 1972: decidability of reversibility and surjectivity.
- 1991: efficient quadratic time algorithm, automata theory.
- J. Kari 1990: undecidable in dimensions 2 and higher.

Surjectivity is injectivity on finite configurations, so these are local properties of phase space.

Bad News

Stronger classifications along the lines of Wolfram's Classes are quite hopeless.

Theorem. *It is Π_2-complete to test if all orbits end in fixed points.*

Theorem. *It is Σ_3-complete to test if all orbits are decidable.*

Theorem. *It is Σ_4-complete to test if a CA is computationally universal.*

The Core Problem

For infinite grids the Reachability Problem

$$x \xrightarrow{*} y$$

is undecidable. Likewise, the closely related Confluence Problem (leading to the same limit cycle) is undecidable

$$\exists\, z\, (x \xrightarrow{*} z \wedge y \xrightarrow{*} z)$$

Theorem. *The Reachability Problem and the Confluence Problem simultaneously can have arbitrary recursively enumerable degree of complexity.*

A Caveat

All these results use classical computability theory and consider only

$$\mathcal{C}_{\mathrm{fin}} = \text{all configurations with finite support}$$

Can be extended to other types of configurations with finitary descriptions such as backgrounds, but fails to deal with uncountable spaces.

Perhaps another model of computation would be appropriate.

Type-2 Turing machines are nice, but then equality is undecidable.

Model Checking

Entscheidungsproblem

The Entscheidungsproblem for the 21. Century: shift to computer science.

- Hilbert's Entscheidungsproblem
- Gödel incompleteness
- Presburger arithmetic
- Tarski's quantifier elimination for the reals
- Collin's cylindrical algebra decomposition
- Matiyasevic's undecidability of Diophantine equations

Model Checking

Starting in the 1980's in computer science.

Basic idea: fix some suitable logic and a collection of structures.

Think of a formula φ in the logic as a specification.

The structures could be anything: hardware, software, protocol, . . .

Goal: Verify that the structure conforms to the specification.

The Catch

Slightly different from classical problem:

- structures usually not fixed, and
- focus on efficient algorithms.

$$\mathfrak{A} \models \varphi$$

Enormously important in (commercial) applications.

Algorithmically often very challenging; $\mathfrak{A}$ is often huge and the logic is complicated.

CA as Structures

Discrete dynamical systems, minimalist description. For any local map ρ let

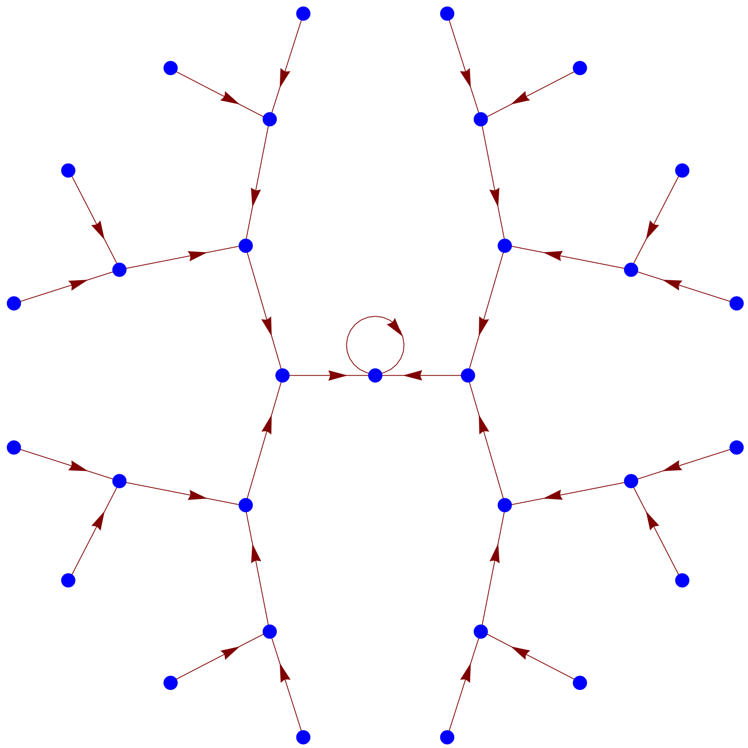
$$\mathfrak{A}_\rho = \langle \mathcal{C}, \to \rangle$$

where, say, $\mathcal{C} = \Sigma^{\mathbb{Z}}$ is the space of *configurations* of the system.

- The definitional properties are summarized in $\mathcal{C}$.
- The global map is given by $\to$, the "next configuration" relation.

Pedestrian Logic

Boundedly many applications of the global map: standard first order logic.



Some Formulae

$$\forall\, x\, \exists\, y\, (y \to x)$$

$$\forall\, x, y, z\, (x \to z \wedge y \to z \Rightarrow x = y)$$

$$\forall\, x\, \exists\, y, z\, (y \to x \wedge z \to x \wedge \forall\, u\, (u \to x \Rightarrow u = y \vee u = z))$$

But these require stronger logic (MSO,TrCL, . . .):

$$\forall\, x\, (x \xrightarrow{*} \mathbf{0})$$

$$\forall\, x\, \exists\, z\, (x \xrightarrow{*} z \wedge z \to z)$$

Model Checking CA

So how do we decide, say, injectivity:

$$\mathfrak{A}_\rho \models \forall\, x, y, z\, (x \to z \wedge y \to z \Rightarrow x = y)$$

We need to deal with

- predicates $x \to y$ and $x = y$,
- boolean connectives $\wedge$, $\vee$, $\neg$ and $\Rightarrow$,
- quantifiers $\forall$ and $\exists$.

Automata

Express $x \to y$ in terms of finite state machines on infinite words, ditto for equality.

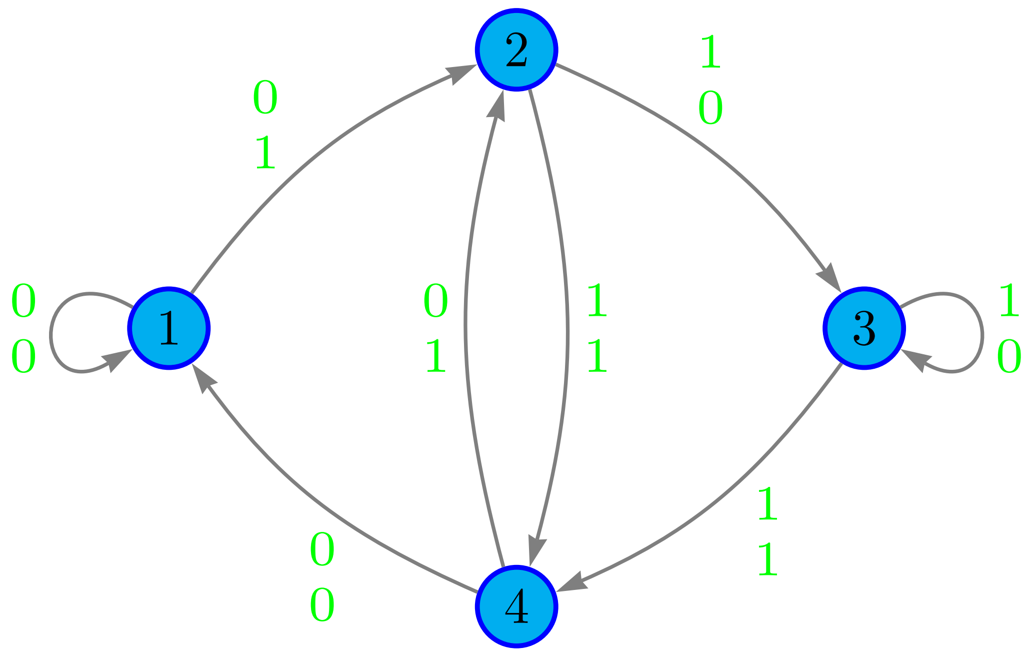
Consider bi-infinite words over Σ^2:

$\ldots$	x_{-3}	x_{-2}	x_{-1}	x_0	x_1	x_2	x_3	$\ldots$
$\ldots$	y_{-3}	y_{-2}	y_{-1}	y_0	y_1	y_2	y_3	$\ldots$

Automata for bi-infinite words:

- a standard semiautomaton $\langle Q, \Gamma, \delta \rangle$,
- an acceptance condition: there is a bi-infinite path in the automaton labeled by the word (a white lie).

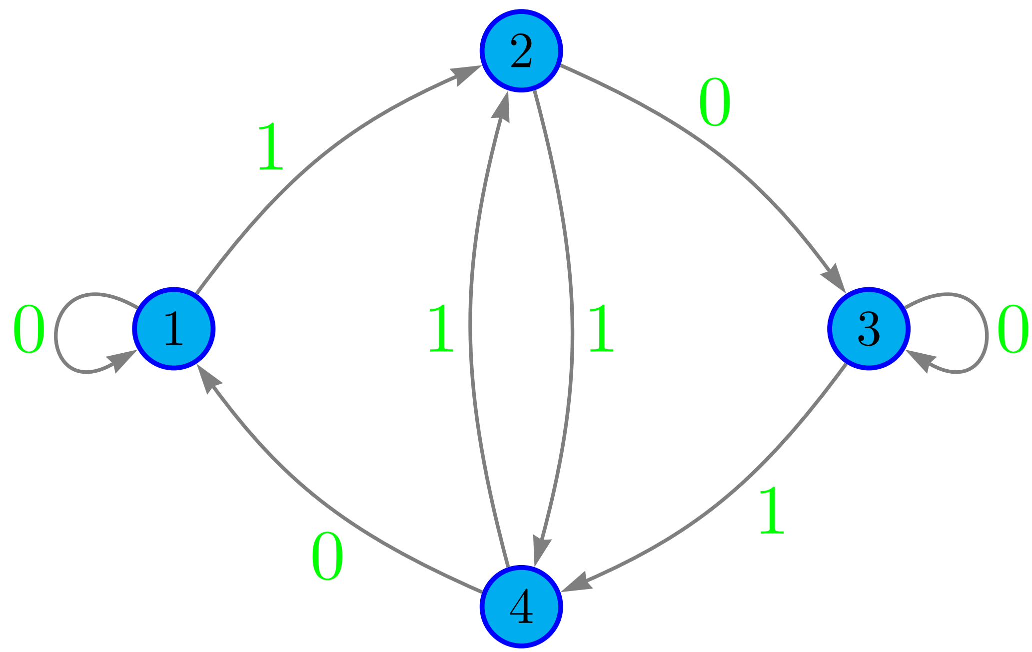
Example: $\mathrm{CA}(2,2,6)$



The canonical automaton $\mathcal{A}_\rho(x,y)$ for the local map $\rho(\boldsymbol{x}) = x_0 \oplus x_1$.

Pleasant Surprise

Existential quantifiers are easy: for $\exists\, x\, \varphi$ just remove the x-track (project on the other tracks.



Universal quantifiers can be rewritten as $\neg\exists\, x\, \neg\varphi$

Boolean Connectives

Only need to deal with $\vee$ and $\neg$.

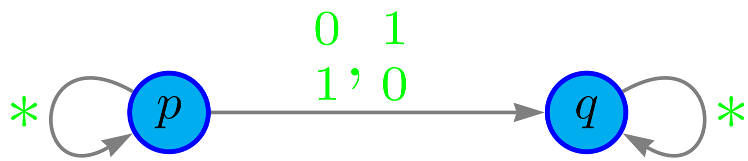
For $\vee$ use a simple disjoint union of the two automata.

Actually, $\wedge$ is easy too: product machine construction.

But $\neg$ causes problems: we need to complement the automaton. Complementation is usually based on determinization. Alas, the simple semiautomata we have so far do not permit determinization.

Non-Equal

How can one test if $x \neq y$?



But now p must be initial (touched infinitely often in the past) and q must be initial (touched infinitely often in the future).

ζ-Automata

Called ζ-automata; essentially two-way Büchi automata.

- Büchi automata operate on one-way infinite words Γ^ω, we have two-way infinite words Γ^∞. Two-way infinite words have no intrinsic coordinates.
- Complementation of ω-automata requires determinization which requires Muller or Rabin automata; the algorithm is a rather intricate and has exponential complexity.
- Complementation of ζ-automata is correspondingly worse.

Logic to Automata

At any rate, for any formula φ we obtain an automaton $\mathcal{A}_\varphi$ such that

$$\mathcal{L}(\mathcal{A}_\varphi) = \{\, (x_1, x_2, \ldots, x_k) \in \Sigma^\infty \mid \mathfrak{A}_\rho \models \varphi(x_1, x_2, \ldots, x_k) \,\}$$

The basic decision problems for these automata are decidable (Emptiness, Universality, Inclusion, Equality).

Decidability

Theorem.
Model checking is decidable for plain FOL $\mathcal{L}(\to, =)$.

But note that the complexity is not elementary: nested complementation is a fiasco efficiency-wise.

$$\text{Büchi} \longrightarrow \text{Rabin} : \qquad 2^{O(n \log n)}$$

Thus one should not expect practical algorithms except for very simple formulae.

The Real Challenge

Is That All?

Can one push the result?

- change the logic
- consider special CAs
- consider special configuration
- like, whatever?

Adding Regular Predicates

Since we are using ζ-automata for the decision algorithm one can add unary predicates ranging over ζ-regular languages.

Example:

${}^{\omega}0\,\Sigma^{\star}\,0^{\omega}$	finite configurations
${}^{\omega}u\,\Sigma^{\star}\,v^{\omega}$	backgrounds

Application: Since a CA is surjective iff it is injective on finite configurations one obtains a quadratic algorithm for injectivity testing.

Adding Nondeterminism

The method works for nondeterministic cellular automata (in fact any kind of ζ-transducer).

Of course, at the low end this may require one more determinization step and thus ruin complexity.

Finite Spaces

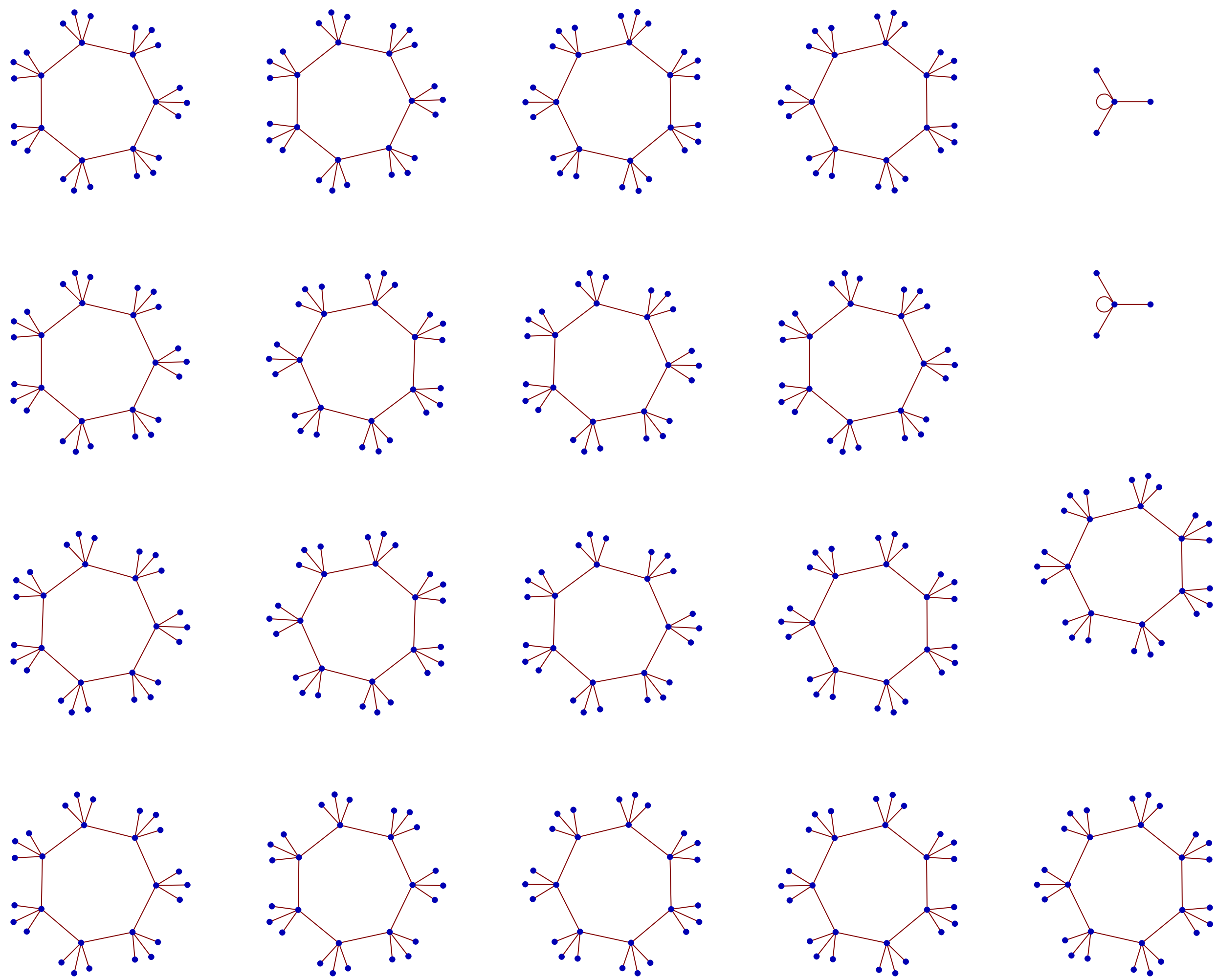
Consider finite configurations spaces:

$$\mathfrak{A}_\rho^n = \langle \mathcal{C}_n, \rightarrow, = \rangle$$

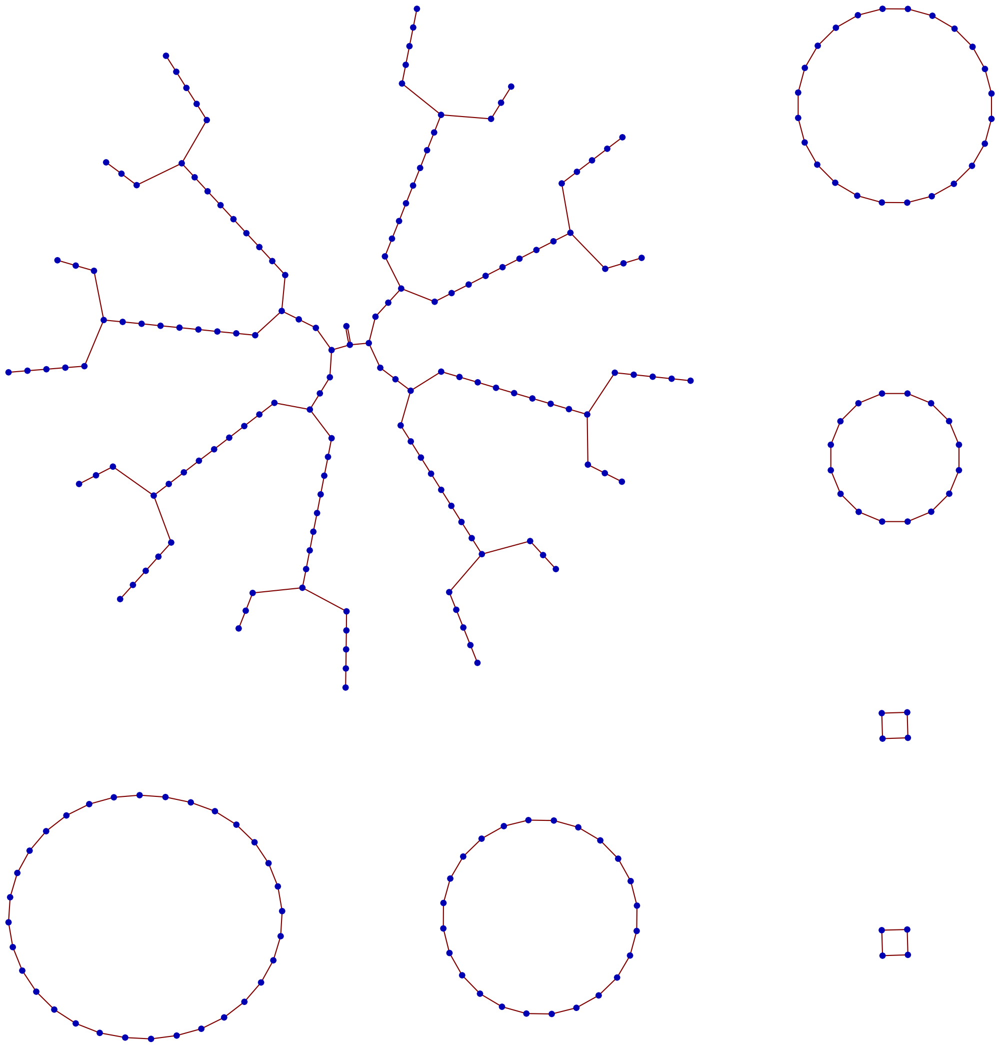
where $\mathcal{C}_n$ is the space of configurations of size n.

So we are looking at nice, finite functional digraphs.

Phase Space



Phase Space



Dire Warning

Nota bene: The grid size n is a free parameter.

The Spectrum

Ideally, given any sentence φ, we would like to understand its spectrum:

$$\mathsf{spec}(\varphi) = \{\, n \in \mathbb{N} \mid \mathfrak{A}_\rho^n \models \varphi \,\}$$

So $\mathsf{spec}(\varphi) = \mathbb{N}$ means "always true", $\mathsf{spec}(\varphi) = \emptyset$ means "always false".

Regular Spectra

ECA 90: $\text{spec}(\text{injective}) = \emptyset$.

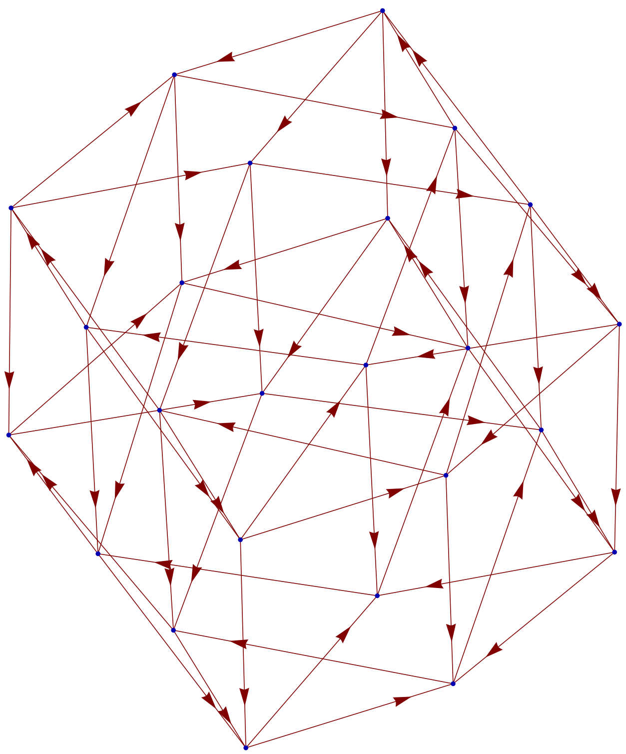
ECA 154: $\text{spec}(\text{injective}) = \text{odd}$.

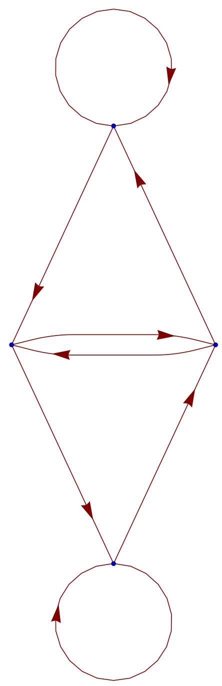
ECA 150: $\text{spec}(\text{injective}) = \mathbb{N} - 3\mathbb{N}$.

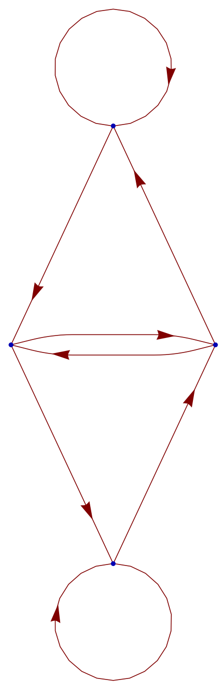
Theorem.
Spectra are regular: the language $\{\, 0^n \mid n \in \text{spec}(\varphi) \,\}$ *is regular.*

Moreover, a corresponding finite automaton can be constructed effectively.

Frivolous Picture







The Usual Spoiler

Theorem.
It is Π_1-complete to test if all configurations on finite grids evolve to a fixed point.

$$\forall\, x\, \exists\, z\, (x \xrightarrow{*} z \wedge z \to z)$$

So dealing with stronger logics is going to be difficult.

Questions

A Typical Question

- fix rule ECA 150
- consider finite grids, cyclic boundary
- logic

$$\mathcal{L}(\rightarrow, \xrightarrow{*}, =)$$

Is this decidable (can we compute the spectrum of a formula)?

More Questions

- What is the complexity of model checking for CA with FOL, at least for simple classes of formulae?
- Is there any interesting logic $\mathcal{L}$ other than FOL with decidable model checking?
- How about subclasses of CA, in particular linear CA?
- How about simple CA on finite grids? Say additive rules?
- When is the spectrum computable?

Even More Questions

The theory of a structure is the collection of all true sentences over that structure.

- Is $\mathrm{Th}(\mathfrak{A}_\rho)$ a useful measure of the complexity?
- In particular for logics stronger than just $\mathcal{L}(\to, =)$?
- Is the theory of Wolfram Class III the same as Class IV?