Rice-Like Theorems for Automata Networks

Guilhem Gamard
Aix-Marseille Université, Université de Toulon, CNRS, LIS, Marseille, France

Pierre Guillon
Aix-Marseille Université, CNRS, I2M, Marseille, France

Kevin Perrot
Aix-Marseille Université, Université de Toulon, CNRS, LIS, Marseille, France
Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France

Guillaume Theyssier
Aix-Marseille Université, CNRS, I2M, Marseille, France

Abstract

We prove general complexity lower bounds on automata networks, in the style of Rice's theorem, but in the computable world. Our main result is that testing any fixed first-order property on the dynamics of an automata network is either trivial, or NP-hard, or coNP-hard. Moreover, there exist such properties that are arbitrarily high in the polynomial-time hierarchy. We also prove that testing a first-order property given as input on an automata network (also part of the input) is PSPACE-hard. Besides, we show that, under a natural effectiveness condition, any nontrivial property of the limit set of a nondeterministic network is PSPACE-hard. We also show that it is PSPACE-hard to separate deterministic networks with a very high and a very low number of limit configurations; however, the problem of deciding whether the number of limit configurations is maximal up to a polynomial quantity belongs to the polynomial-time hierarchy.

2012 ACM Subject Classification Theory of computation → Models of computation; Theory of computation → Complexity classes

Keywords and phrases Automata networks, Rice theorem, complexity classes, polynomial hierarchy, hardness

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.32

Funding This work has been partially supported by the French ANR project FANs ANR-18-CE40-0002.

Acknowledgements We would like to thank the colleagues that were involved in some discussions on the topic at the early stages of this work and somehow convinced us to pursue and finally settle the main result.

1 Introduction

An automata network is a digraph where each node holds a state (among a finite set) that evolves in function of the states of its inbound neighbors. All the nodes evolve at the same time, in parallel. In other terms, the main difference between an automata network and a cellular automaton is that the "grid" may be an arbitrary finite digraph, and that different cells (nodes) may have different local functions. Since this definition is very general, any finite dynamical system may be encoded into an automata network in a reasonable fashion.

Initially, *Boolean* automata networks, where the set of states is required to be $\{0,1\}$ for all nodes, were introduced in the 1940's as a formal model of neural networks [16]. Subsequently, *linear* automata networks, where the evolution function of each node is a linear combination of its inputs, were investigated [3, 6, 9, 15], still motivated by neural networks. General automata networks were then introduced in theoretical biology, in order to study the

38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021).
Editors: Markus Bläser and Benjamin Monmege; Article No. 32; pp. 32:1–32:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

dynamics of gene expression and inhibition [14, 24]. They have since been further considered, mostly from the standpoint of applications [7, 13, 17, 23], although theoretical results also appeared [1, 4, 8, 19, 20].

In the literature, many questions about automata networks deal with the *dynamics* of the system, i.e., the global function that it computes. For instance: does a given network have a fixed point (i.e. a stable configuration)? How many of them does it have? Does it have a cycle of exactly two configurations evolving one to the other? Does it have a configuration with at least three predecessors? As one may suspect, such questions are computationally hard to solve in general. The reason why one may have this intuition is that automata networks can be viewed as a model of computation, so they probably are subject to some kind of theorem in the flavor of Rice's [18]:

▶ **Theorem 1.1.** *Any nontrivial property of the function computed by a Turing machine is undecidable.*

One may object that automata networks are strictly less powerful than Turing machines, for they lack unbounded memory. Any question about the function computed by an automata network may be answered by exhaustive search, i.e., by enumerating all possible configurations of the network (among finitely many), and testing each of them for the desired property. That objection stands but the brute-force approach is not practical, for the number of configurations is exponential in the size of network. On the other hand, most applications of automata networks amount to answering questions about the functions that they compute. We therefore endeavor to prove results along the lines of [2]:

▶ **Metatheorem 1.2.** *Any nontrivial property of the function computed by an automata network has high computational complexity.*

Typically, "high computational complexity" means something like "NP-hard", "co-NP-hard" or even "PSPACE-hard". As a consequence, there is probably no approach significantly faster than brute-force for those questions, which makes them out of reach for our current computational power. Thus, our results show that any application of automata networks requiring fast testing of some dynamical property will have to rely on specific aspects of the practical situation under consideration.

In order to make the statement of the metatheorem precise, we need to specify the concepts of "property" and "nontriviality". We obtained several different results that fit the pattern of Metatheorem 1.2, with various tradeoffs on "property" and "nontriviality", as explained in the *contributions and organization of the paper* paragraph below.

Let F denote an automata network and X its set of configurations. The *dynamics* of F may refer to two equivalent objects: either the function $f : X \to X$ given by the action of F; or the *transition digraph* (X, E) where E is the set $\{(x, f(x)) \mid x \in X\}$.

Specifying an automata network – say, by giving a Boolean circuit for the local function of each node – is a way to specify its transition digraph in a concise way. Thus, some of our results may be interpreted as statements about succinct graphs. However, in this paper we mostly consider *deterministic* automata networks, i.e., networks whose transition digraph has out-degree one. This restriction is not common in graph theory nor in finite model theory. Still, some generalizations of our results might be of interest for those communities.

As usual with dynamical systems, the *long-term behavior* of an automata network is of special interest. For pumping reasons, a deterministic automata network is always ultimately periodic. Many practical questions can be asked about both the transient and the periodic regimes of the system; therefore, it is interesting to know that such questions are computationally hard. The periodic regime of an automata network is called its *limit dynamics.* It is the dynamics spanned by the configurations that are always visited infinitely many times whenever they are visited once. We have two kinds of results: some are about the limit dynamics of a given automata network, and some are about the full dynamics.

Contributions and organization of the paper

- In Section 2, we set up the formalism: definitions, first remarks, etc.
- In Section 3, we prove that a large class of properties over the set of limit configurations of networks are PSPACE-complete. This echoes a result from [12] on cellular automata.
- In Section 4, we show that it is PSPACE-complete to distinguish automata networks with a very small and a very high number of limit configurations. However, we show that the problem of deciding whether this number is maximal up to a polynomial quantity (in the number of states and the number of nodes) belongs to the arithmetical hierarchy.
- In Section 5, we prove that if a property on the dynamics of automata networks is expressible by a first-order formula over a simple signature, then its complexity is either bounded, NP-hard, or co-NP-hard. In this setting, the formula is considered fixed, not part of the input. We also observe that this result still holds when restricted to bijective automata networks, or to limit dynamics instead of full dynamics.
- In Section 6, we show that if the first-order formula is considered as part of the input, then the previous problem becomes PSPACE-complete.

2 Definitions and terminology

Let $\{A_i\}_{i\in I}$ denote a finite family of finite sets and $A = \prod_{i\in I} A_i$. An *automata network* (AN) is a function f from A to itself. We think of it as a system of finite automata linked to each other, where the input of one automaton is the current state of the other automata (thus there is no external input word). More precisely, A_i is the set of states of the i^{th} automaton (or node); an element of A is a *configuration* of the system (it assigns one state to each automaton); and $f : A \to A$ gives the *evolution* of the system after one step of time.

We can split f into a family of *local functions* $\{f_i\}_{i\in I}$, where f_i goes from A to A_i and returns the state of the i^{th} automaton at the next step. In other terms, if $I = \{1, \dots, n\}$ and $a = (a_1, \dots, a_n)$ is an element of A, then we have $f(a) = (f_1(a), f_2(a), \dots, f_n(a))$. For a given i, it might happen that $f_i(a)$ does not depend on all the components of a. For instance, $f_i(a)$ might depend only on a_i and a_{i-1} (where a_0 stands for a_n). The *interaction digraph* G_f of f is the graph $(I, \mathcal{I})$ where $\mathcal{I}$ is the set of pairs (i, j) such that, for some a, b with $a_k = b_k$ for every $k \neq i$, we have $f_j(a) \neq f_j(b)$. A configuration of an automata network may be viewed as a labeling of the interaction digraph. The label of each node evolves under f, but the new label depends only on the labels of the inbound neighbours of the node.

The *dynamics*, or *transition digraph*, of a network f, denoted by $\mathcal{G}_f$, is the graph of the function f: it is given by $(A, \mathcal{F})$, where $\mathcal{F}$ is the set of pairs $(a, f(a))$, for a ranging over A. The *limit set* of an automata network f is denoted Ω_f and defined as $\Omega_f = \bigcap_{n\in\mathbb{N}} f^n(A)$. Its elements are the *limit configurations*, which are those that are met infinitely often in at least one execution of the system. The *limit dynamics* or the *limit digraph* of f is denoted $\mathcal{G}_f^\omega$ and is the subgraph of $\mathcal{G}_f$ induced by Ω_f. Figure 1 illustrates the definitions so far.

When all the A_i's are equal, we say that f is an *automata network with uniform alphabet,* or ANU for short.

All these definitions generalize immediately if f is a relation instead of a function; the f_i's are then *local relations.* Such an object is called a *nondeterministic* automata network. Unless explicitly mentioned, automata networks are supposed to be deterministic.

$$f_1(x) = \begin{cases} x_1 + 1 & \text{if } x_1 \neq 2 \\ x_1 - 1 & \text{otherwise} \end{cases}$$

$$f_2(x) = \begin{cases} x_2 + 1 & \text{if } x_1 \neq 2 \\ x_2 - 1 & \text{otherwise} \end{cases}$$

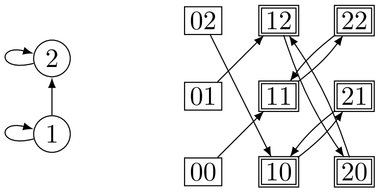

Figure 1 Example of automata network f for $A_1 = A_2 = \{0, 1, 2\}$ (left), its interaction digraph G_f (middle), and its transition digraph $\mathcal{G}_f$ (right). $\Omega_f = \{10, 11, 12, 20, 21, 22\}$.

When we need to give an automata network as input for an algorithm, we provide the interaction digraph and one Boolean circuit for each local function (or relation). Circuit sizes are assumed to be at most $|A_i|^{|A|}$ (or $2^{|A_i| \cdot |A|}$), because those are the sizes of the corresponding truth tables. The nodes of the interaction digraph are assumed to be numbered 1, …, n and each A_i is assumed to be of the form $\{0, \ldots, |A_i| - 1\}$.

If P is a property that automata networks may or may not satisfy, and f is an automata network, then we write $f \models P$ if f satisfies P, and $f \not\models P$ otherwise. This is an abuse of notation, and its precise meaning depends on the exact nature of P.

Unless otherwise stated, our reductions are polynomial-time many-one, and $\leq_{tt}^p$ denotes polynomial-time truth-table reduction. For every integer k, the symbol Σ_k^p denotes the level Σ_k of the polynomial hierarchy. For a decision problem P where an ANU is given as input, it is natural to consider the Q-variant where the inputs are restricted to ANU having alphabet Q. In the following, we will say that P is hard *with fixed alphabet* if there exists some Q such that the Q-variant of the problem is hard. It will be the case of most of our hardness results on ANU.

3 Abstract properties of limit sets

In this section, we focus on properties of the *limit set* and establish a Rice-like theorem similar to the well-known result for limit sets of cellular automata [12].

A property P is a *limit set property* if, whenever two AN f and g have the same limit sets ($\Omega_f = \Omega_g$), the following holds: $f \models P \iff g \models P$. The simplest possible limit set is a singleton, and the following lemma already shows that separating between singleton limit sets and exponentially large ones can depend on arbitrary linear space Turing computations.

▶ **Lemma 3.1.** *For any Turing machine M, any $k \in \mathbb{N}$, any $n \in \mathbb{N}$ and any input u for M of size at most n, there is an alphabet Q depending only on M and k and a deterministic ANU $f_{M,k,n,u} : Q^n \to Q^n$ and $q_0 \in Q$ such that:*

- *if M accepts input u in at most $k^n - 1$ steps, using space at most n, then $\Omega_{f_{M,k,n,u}} = \{q_0^n\}$;*
- *otherwise, there is a configuration $c \in \Omega_{f_{M,k,n,u}}$ belonging to a cyclic orbit of length k^n where state q_0 never appears: $\forall t \in \mathbb{N}, \forall i \in \{1, \ldots, n\} : f_{M,k,n,u}^t(c)_i \neq q_0$.*

Moreover, circuits for the local functions of $f_{M,k,n,u}$ can be computed in time $\mathsf{poly}(M, k, n, u)$.

Proof. Let Σ be the alphabet, S the state set of M, and $\perp$, q_0 fresh symbols. Define $H = S \sqcup \{\perp\}$ and $Q = (\Sigma \times H \times \{0, \ldots, k-1\}) \sqcup \{q_0\}$. Any configuration $c \in Q^n$ not containing state q_0 (*i.e.* $c_i \neq q_0$ for all i) can be seen as a triple (c^Σ, c^H, c^k) of configurations in Σ^n, H^n and $\{1, \ldots, k\}^n$ respectively. We say that configuration c is *valid* if it does not contain state q_0 and there is exactly one position i such that $c_i^H \in S$. If c is valid, c^Σ encodes the content of the tape (limited to n cells), c^H encodes the position and the state of the

Turing head, and c^k encodes a counter between 0 and $k^n - 1$ in base k. We call c_0 the valid configuration where c^Σ represents the tape containing input u starting on the leftmost position of the (finite) tape, c^H represents the head in the initial state on the leftmost position of the tape, and c^k represents the number 0. The behavior of $f_{M,k,n,u}$ is as follows:

1. send any invalid configuration to q_0^n;
2. send any valid configuration c such that c^k represents value $k^n - 1$ to configuration c_0;
3. send any valid c such that c^H represents a head in an accepting state to q_0^n;
4. for any valid configuration c from which one step of M does not make the head move outside the n-cell tape, $f_{M,k,n,u}$ performs this step and increments the value in c^k;
5. for any other valid configuration c, leave c^Σ and c^H unchanged but increment c^k.

It should be clear enough from the above description that circuits computing local maps of $f_{M,k,n,u}$ can be produced in polynomial time given M, k, n and u.

Suppose first that M halts on input u in at most $k^n - 1$ steps and using space at most n and suppose for the sake of contradiction that there is a configuration $c \in \Omega_{f_{M,k,n,u}}$ which is not q_0^n. Then q_0^n cannot be in the orbit of c, so only cases 2, 4 and 5 are used in the orbit of c. The counter is always incremented, until reaching $k^n - 1$, so that c_0 must appear in the (periodic) orbit of c, and therefore c is in the orbit of c_0. We get a contradiction because M halts on input u in at most $k^n - 1$ steps and using space at most n, so that case 3 must be triggered at the corresponding step in the orbit of c_0.

Suppose now that M does not halt within time $k^n - 1$ and space n starting from input u. We claim that c_0 belongs to a cyclic orbit of length k^n and that state q_0 cannot appear in this orbit. Indeed, validity of configurations is preserved under iteration of $f_{M,k,n,u}$ except in case 3, which is discarded by hypothesis, and after $k^n - 1$ applications of case 4 or 5, during which the counter component c^k is constantly incremented, we reach case 2 and the orbits cycles back to c_0. ◀

▶ **Corollary 3.2.** *The following problem is* PSPACE*-complete, even with fixed alphabet:*

Nilpotency
Input: a deterministic ANU $f : \{0, \dots, q-1\}^n \to \{0, \dots, q-1\}^n$.
Question: does $|\Omega_f| = 1$*?*

Proof. First, the problem is in PSPACE because checking that an AN on $\{q\}^n$ has a singleton limit set can be done by checking that $f^{q^n}(x)$ is the same configuration for all $x \in \{q\}^n$. Second, we can make a reduction from quantified Boolean satisfiability (QBF) problem [22] as follows: let M be any Turing machine that solves the QBF problem in linear space by a brute force algorithm and let k be large enough so that M works in less than k^n time steps on instances of QBF of size at most n. By Lemma 3.1, given an instance u of size at most n of QBF to be solved by M, the AN $f_{M,k,n,u}$ can be produced in polynomial time and has a singleton limit set if and only if u is true. The alphabet of $f_{M,k,n,u}$ only depends on machine M, so we have a reduction working with fixed alphabet ANU. ◀

Next, we present another theorem, whose proof is inspired by [12]. Intuitively, the firing squad from [12] is replaced by nondeterminism, and the nilpotency problem is replaced by the problem of having an orbit completely avoiding a given state (whose hardness is established by Lemma 3.1).

Given a collection of AN (deterministic or not, ANU or not, etc), we say that a property is *effectively nontrivial* in the collection if there is a polynomial-time algorithm that, given n in unary, produces two AN with n nodes belonging in this collection, one that satisfies the property and another one that does not. This condition of effectiveness is natural since, if one wants to make some reduction to prove that a property is hard, then the reduction usually induces an algorithm to produce models and counter-models of the property.

▶ **Theorem 3.3.** *Effectively nontrivial limit set properties of nondeterministic AN are the same as effectively nontrivial limit set properties of nondeterministic ANU. If $\mathcal{P}$ is an effectively nontrivial limit set property of nondeterministic ANU, then the following problem is* PSPACE*-hard for the $\leq_{tt}^{p}$ reduction:*

$\mathcal{P}$-limit-set
Input: a nondeterministic ANU $f : \{0,\dots,q-1\}^n \to \{0,\dots,q-1\}^n$.
Question: does $f \models \mathcal{P}$?

Proof. Every effectively nontrivial limit set property for nondeterministic ANU is also effectively nontrivial for nondeterministic AN. Conversely, if P is effectively nontrivial for nondeterministic AN, then there is a polynomial-time algorithm which, given n, produces a model f_1 and a counter-model f_2 of P that may have nonuniform alphabets, but we can extend them to larger alphabets while preserving the limit set by sending (deterministically) any extra configuration to a fixed one that uses only the original alphabet. This transformation is effective (a description by circuits of the new rule can be computed in polynomial time from the description of the original rule). This proves the first claim of the theorem and allows us to focus on ANU.

Given two (possibly nondeterministic) ANU f_1 and f_2 both acting on Q_f^n, and a deterministic one h on Q_h^n with some distinguished state $q_0 \in Q_h$, we define two nondeterministic ANU g_1 and g_2 both acting on $(Q_f \cup (Q_f \times Q_h))^n$ as follows. We fix some $q \in Q_f$. Intuitively, g_i mimics f_i on Q_f^n, and either mimics h on $(Q_f \times Q_h)^n$ or projects onto the Q_f component provided state q_0 is not present in the Q_h component of states. In any other case, the behavior is go to configuration q^n deterministically. To simplify notation we see any configuration $x \in (Q_f \times Q_h)^n$ as a pair $x = (x^f, x^h) \in Q_f^n \times Q_h^n$. Then g_i is defined as follows:

$$g_i(x)_v = \begin{cases} f_i(x)_v & \text{if } x \in Q_f^n, \\ \{(x_v^f, h(x^h)_v), x_v^f\} & \text{if } x = (x^f, x^h) \in Q_f^n \times Q_h^n \text{ and } x_j^h \neq q_0 \text{ for all } j \in [n], \\ q & \text{otherwise,} \end{cases}$$

for $v \in \{n\}$ and $i = 1, 2$. It is straightforward to check that if state q_0 appears in all orbits of h, then $\Omega_{g_i} = \Omega_{f_i}$ because, in this case, any orbit of g_i must end up in Q_f^n. In particular, in this case, $\Omega_{g_1} \neq \Omega_{g_2}$ because f_1 and f_2 are respectively a model and a counter-model of the limit-set property P. On the other hand, if h has some orbit that completely avoids state q_0, then for any $x \in Q_f^n$ we have $x \in \Omega_{g_i}$ because x can be reached arbitrarily late from (x, y) in the dynamics of g_i where y is any configuration of the considered orbit of h. Moreover in this case $\Omega_{g_1} = \Omega_{g_2}$ holds because, by definition, for any $y \notin Q_f^n$ we have $y \in \Omega_{g_1} \Leftrightarrow y \in \Omega_{g_2}$. Thus we have a $\leq_{tt}^{p}$ reduction from the problem of deciding whether h has an orbit completely avoiding state q_0 to the property P: the former can be decided by checking whether $\Omega_{g_1} = \Omega_{g_2}$.

To conclude the proof, it is sufficient to invoke Lemma 3.1 and use an argument similar to the proof of Corollary 3.2, in order to show that deciding whether a given AN h has an orbit completely avoiding state q_0 is PSPACE-hard. ◀

4 Size of limit sets

In this section, we are interested in problems about the size of limit sets. First, if we take the settings of nondeterministic AN as in the previous section, Theorem 3.3 already tells us that any effectively nontrivial problem about the size of the limit set will be PSPACE-hard, as a particular limit set property. We now focus on deterministic ANU, and the following canonical problems on the size of limit sets.

Given a map $\lambda : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that $\lambda(q, n) \leq q^n$, we define the problem P_λ as follows:

Problem P_λ
Input: a deterministic ANU $f : \{0, \dots, q-1\}^n \to \{0, \dots, q-1\}^n$.
Question: does $|\Omega_f| \geq \lambda(q, n)$?

The goal of this section is to show that problem P_λ jumps from PSPACE-hardness down to the polynomial-time hierarchy depending on λ. First, when λ stays far enough from the total number of configurations, we already have the tools to conclude PSPACE-hardness.

Using Lemma 3.1 as in the proof of Corollary 3.2 we obtain the following theorem.

▶ **Theorem 4.1.** *Let $\lambda : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a map such that for some $k > 1$ and for any $n \in \mathbb{N}$ it holds $2 \leq \lambda(q, n) \leq k^n$. Then the problem P_λ is PSPACE-hard, even with fixed alphabet.*

However, the problem whether the size of the limit set is maximal up to a polynomial quantity belongs in the polynomial-time hierarchy. The intuition is that if the limit set is close to maximal, then it is reached quickly under iterations of the AN.

▶ **Proposition 4.2.** *Let $\delta : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a polynomial map, and define $\lambda(n, q) = q^n - \delta(n, q)$. Then problem P_λ is Σ_3^p and co-NP-hard, even with fixed alphabet.*

Proof. Let us denote $[q] = \{0, \dots, q-1\}$. Consider any deterministic ANU $f : [q]^n \to [q]^n$. We claim that if $|\Omega_f| \geq \lambda(q, n)$ then $\Omega_f = f^{\delta(q,n)}([q]^n) = f^{\delta(q,n)+1}([q]^n)$. Indeed, by induction, we have that $f^k([q]^n) = f^{k+1}([q]^n)$ for some k implies that $f^k([q]^n) = \Omega_f$. Therefore $\Omega_f \subsetneq f^{\delta(q,n)}([q]^n)$ would imply $|\Omega_f| < q^n - \delta(q, n)$. The claim follows. Conversely, the same argument shows that $f^{\delta(q,n)}([q]^n) \neq f^{\delta(q,n)+1}([q]^n)$ implies $|\Omega_f| < \lambda(q, n)$.

We deduce that the problem P_λ is equivalent to: "$f^{\delta(q,n)}([q]^n) = f^{\delta(q,n)+1}([q]^n)$ and $|[q]^n \setminus f^{\delta(q,n)}([q]^n)| \leq \delta(q, n)$." This can be rephrased as the conjunction:

- $\forall x \in [q]^n, \exists y \in [q]^n$ such that $f^{\delta(q,n)}(x) = f^{\delta(q,n)+1}(y)$, and
- there is a set $L \subseteq [q]^n$ of $\delta(q, n)$ distinct configurations such that $\forall x \in [q]^n, f^{\delta(q,n)}(x) \notin L$ and $\forall x \in [q]^n, x \notin L \Rightarrow \exists y \in [q]^n, f^{\delta(q,n)}(y) = x$

This shows that the problem P_λ is Σ_3^p.

To show co-NP-hardness of problem P_λ we make a reduction from UNSAT. First note that $\delta(4, n) < 2^n - 1$ for large enough n because $\delta(4, n)$ is a polynomial in n. Then, given any instance ϕ of UNSAT with n variables, build the ANU $f : [4]^n \to [4]^n$ as follows:

$$f(x) = \begin{cases} 0^n & \text{if } \pi(x) \text{ represents a statisfying assignment of variables for } \phi, \\ x & \text{otherwise.} \end{cases}$$

where $\pi(x_1, \dots, x_n) = (t_1, \dots, t_n)$ with t_i true if and only if $x_i = 0 \bmod 2$. It is clear that for any assignment of variables $(t_1, \dots, t_n)$ there are 2^n possible choices of x such that $\pi(x) = (t_1, \dots, t_n)$. Moreover, if $x \neq 0^n$, then $x \notin \Omega_f$ when $\pi(x)$ is a satisfying assignment for ϕ. Therefore if ϕ is satisfiable then $|\Omega_f| \leq 4^n - 2^n + 1 < \lambda(4, n)$. On the contrary, if ϕ is not satisfiable, then $f(x) = x$ for all $x \in [4]^n$ so we have $|\Omega_f| \geq \lambda(4, n)$. co-NP-hardness of problem P_λ follows. ◀

5 First-order properties of transition digraphs are hard

In this section, a *graph* is the transition digraph of some deterministic automata network, i.e., a simple digraph where all vertices have out-degree 1 and where self-loops are allowed. A *formula* means a closed first-order logic formula over the signature $\{=, \to\}$ (binary relations). Formulas will be evaluated in graphs, so "$\forall x$" is understood as "for all vertex x" and "$x \to y$" is understood as "there is an edge from x to y". For all formula ψ, define:

ψ-Dynamics
Input: an automata network f.
Question: does $\mathcal{G}_f \models \psi$?

Note that the formula ψ is *not* part of the input, but rather a parameter of the problem.

▶ **Definition 5.1.** *A formula ψ is ω-nontrivial if there are infinitely many models and infinitely many countermodels.*

▶ **Theorem 5.2.** *If ψ is ω-nontrivial, then ψ-Dynamics is either* NP- *or* co-NP-*hard.*

The condition of ω-nontriviality is optimal: indeed, if ψ is ω-trivial, then solving ψ-Dynamics amounts to testing whether the given AN belongs to a finite fixed list of objects, which can be done in time $O(1)$. (Recall from Section 2 that the circuits sizes are bounded by $|A_i|^{|A|}$.) Whether the problem is NP- or co-NP-hard varies with ψ. The proof consists of the next three subsections.

5.1 Encoding SAT instances into the dynamics of AN

The results in this subsection provide a general tool to deduce hardness from pumping constructions. We recommend that first-time readers skip the definitions of $\sqcup_2$ and $\sqcup_3$ and fix $z = 1$ everywhere, because the cases $z = 2, 3$ will not be needed until much later in the paper. In the next definition, "pointed" nodes are simply distinguished vertices in a graph.

▶ **Definition 5.1.1.** Let G, G' denote graphs; we define three operators $\sqcup_1, \sqcup_2, \sqcup_3$.

- The graph $G \sqcup_1 G'$ (or $G \sqcup G'$) is the disjoint union of a copy of G and a copy of G'.
- If G has a pointed node v and G' has any number of pointed nodes (possibly zero), then the graph $G \sqcup_2 G'$ is $G \sqcup_1 G'$ except that each edge going out of a pointed node of G' points to v instead. The result has one pointed node, v.
- If G has a pair of pointed nodes (u, v) and G' has a pair of pointed nodes (u', v'), then $G \sqcup_3 G'$ is $G \sqcup_1 G'$ except that: the edge going out of v points to u'; and the edge going out of v' points to u. Besides, $G \sqcup_3 G'$ has pointed nodes (u', v).

If G is a graph, k is an integer, and z is in $\{1, 2, 3\}$, then $\bigsqcup_z^k G$ denotes $G \sqcup_z \cdots \sqcup_z G$, with k copies of G. Now let n be an integer, $\Gamma = (G_1, \dots, G_n)$ a n-tuple of graphs, and w a word over alphabet $\{1, \dots, n\}$. Define $\mathcal{U}_z^{G,\Gamma}(w)$ by induction as follows: $\mathcal{U}_z^{G,\Gamma}(\varepsilon) = G$, and $\mathcal{U}_z^{G,\Gamma}(w_1 \dots w_k) = \mathcal{U}_z^{G,\Gamma}(w_1 \dots w_{k-1}) \sqcup_z G_{w_k}$ (where ε is the empty word). See Figure 2.

▶ **Proposition 5.1.2.** *Let ψ be a formula and z be an element of $\{1, 2, 3\}$. If there exist nonempty graphs G, J, D such that for all integers k and k', we have $G \sqcup_z (\bigsqcup_z^k J) \not\models \psi$ and $G \sqcup_z (\bigsqcup_z^k J) \sqcup_z (\bigsqcup_z^{k'} D) \models \psi$, then ψ-Dynamics is* NP-*hard.*

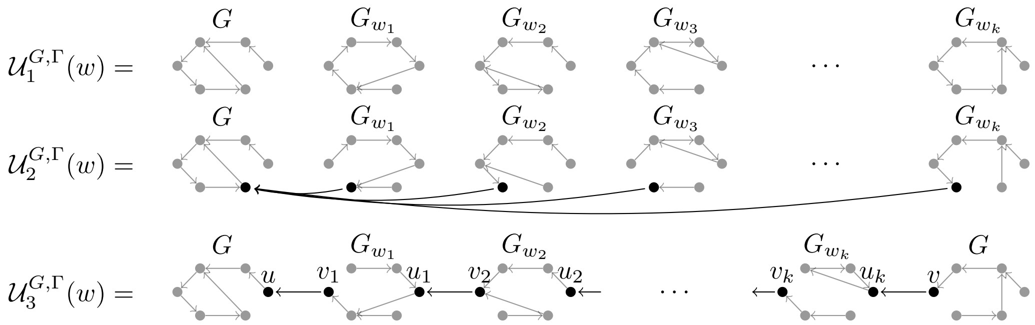
$\mathcal{U}_1^{G,\Gamma}(w) =$
$\mathcal{U}_2^{G,\Gamma}(w) =$
$\mathcal{U}_3^{G,\Gamma}(w) =$
G
G_{w_1}
G_{w_2}
G_{w_3}
G_{w_k}
$\cdots$
u
v_1
u_1
v_2
u_2
v_k
u_k
v

Figure 2 Illustration of $\mathcal{U}_z^{G,\Gamma}(w)$. In the illustration of $\mathcal{U}_3^{G,\Gamma}(w)$, G is not connected.

▶ **Definition 5.1.3.** Let S denote an instance of SAT with s variables. Then $\overline{S}$ is the word of length 2^s over alphabet $\{1, 2\}$ whose i^{th} letter is 1 if $S(i)$ is false, and 2 if it is true (viewing the binary expansion of i as an assignment for S).

▶ **Lemma 5.1.4.** *Let S be an instance of SAT, $z \in \{1, 2, 3\}$ and G, J, D be graphs such that $1 < |G| < |J| = |D|$. There are an AN f and an integer k such that $\mathcal{G}_f = \mathcal{U}_z^{G,(J,D)}(\overline{S}) \sqcup_z (\bigsqcup_z^k J)$. Moreover, f is computable in polynomial time from S if G, J, D are constant.*

Proof. Let $\delta = \gcd(|G|, |J|)$, and write $|G| = g \cdot \delta$ and $|J| = j \cdot \delta$ for some coprime integers g, j. Call s the number of variables in S. First, find an integer t such that $g \leq 2^t$ and $\gcd(s + t, \varphi(j)) = 1$, where φ denotes Euler's totient. To do so, let $t' = s/\gcd(s, \varphi(j))$, so that t' and $\varphi(j)$ have no common prime factors. Then let t'' denote a power of t' that exceeds $s + \lceil \log_2 g \rceil$ (compute it by successive squarings). Finally, take $t = t'' - s$. Since $\gcd(g, j) = 1$, we can use Algorithm 17.1 of [21] to find an integer $x \geq 1$ such that $x^{s+t} \equiv g \mod j$. For the rest of the proof, assume that $x \geq 2$: indeed, if $x = 1$, then $g \equiv 1 \mod j$, so we can choose $x = g^{\varphi(j)}$ instead by Euler's formula. Since Algorithm 17.1 runs in polynomial time and g, j are constants, we can find x and t in polynomial time.

Assume that $V(G) = \{0, \ldots, |G| - 1\}$ and $V(J) = V(D) = \{0, \ldots, |J| - 1\}$ (recall that $|J| = |D|$). For all relevant integer n, write $G(n)$ (resp. $J(n)$, $D(n)$) the unique successor of n in G (resp. J, D). The automata network f has $1 + s + t$ nodes: one node with alphabet $\{0, \ldots, \delta - 1\}$ and $s + t$ nodes with alphabet $\{0, \ldots, x - 1\}$. It reads its current configuration as an integer N (with $0 \leq N \leq \delta \cdot x^{s+t} - 1$) and transitions as follows:

- If $N < |G|$, then $f(N) = G(N)$.
- If $0 \leq N - |G| < 2^s \cdot |J|$, then by Euclidean division let q, r be the integers such that $N - |G| = |J| \cdot q + r$ and $0 \leq r < |J|$. View q in binary as a valuation for S.
  If $S(q)$ is true, then $f(N) = |J| \cdot q + D(r)$. If $S(q)$ is false, then $f(N) = |J| \cdot q + J(r)$.
- If $2^s \cdot |J| \leq N - |G|$, then let q, r be as in the previous case, and $f(N) = |J| \cdot q + J(r)$.

If $z = 1$, the description of f is complete. If $z = 2$, the pointed nodes of each copy of J and D transition to the pointed node of G instead. If $z = 3$, order all the graphs (G, J's, and D's) according to the configuration number N that encodes their first vertex. Encode the pointed nodes of each graph in their vertices 0 and 1. Make the first pointed node of each graph transition to the second pointed node of the next graph, looping around $\delta \cdot x^{s+t}$.

Since $2 \leq x$, we have $g \leq 2^t \leq x^t$, so one copy of G and at least 2^s copies of J or D fit in the dynamics of f. Besides, since $x^{s+t} \equiv g \mod j$, there are no leftover configurations. The circuits encoding f can be produced in polynomial time: the only part depending on S merely requires to evaluate S. ◀

Proof of Proposition 5.1.2. Let $\widetilde{J} = \bigsqcup_z^{|D|} J$ and $\widetilde{D} = \bigsqcup_z^{|J|} D$, so that $|\widetilde{J}| = |\widetilde{D}|$. The statement follows from Lemma 5.1.4, as the graphs G, $\widetilde{J}$ and $\widetilde{D}$ can be padded with copies of $\widetilde{J}$ to meet the other size constraints. ◀

5.2 From transition digraphs to disjoint unions of labeled cycles

Recall that all our graphs have out-degree 1, so each connected component of a graph is a cycle, in which each vertex is the root of an upward tree (a rooted tree where arcs point towards the root). Define $\mathcal{T}$ as the set of finite, nonempty upward trees. Any graph may be seen as a multiset of cyclic words over alphabet $\mathcal{T}$.

If G and G' are graphs, we write $G \equiv_m G'$ if and only if they satisfy the same formulas of quantifier rank m. Let $\mathcal{E}_m$ denote the set of equivalence classes of $\equiv_m$ over $\mathcal{T}$.

▶ **Lemma 5.2.1.** *For all m, the set $\mathcal{E}_m$ is finite.*

Proof. Without loss of generality, all formulas are in prenex form (quantifiers are at the beginning). Thus, a formula ϕ is of the form $Q_1x_1 \ldots Q_mx_m \; \phi'(x_1, \ldots, x_m)$, where Q_i belongs to $\{\exists, \forall\}$ for all i and ϕ' is a quantifier-free formula. There are 2^m ways to assign quantifiers to the Q_i's. A quantifier-free formula $\phi'(x_0, \ldots, x_{m-1})$ is a Boolean formula over $2m^2$ variables: "$x_i \to x_j$" and "$x_i = x_j$", for $0 \leq i, j < m$. Two Boolean formulas are equivalent if and only if they have the same truth table. There are 2^{2m^2} possible assignments for the "variables", thus $2^{2^{2m^2}}$ possible truth tables. Consequently, there are at most $2^{m+2^{2m^2}}$ nonequivalent formulas of quantifier rank m. Any structure satisfying (resp. falsifying) a formula has to satisfy (resp. falsify) all formulas equivalent to it. Therefore, there are finitely many possible sets of formulas of quantifier rank m that a given structure may satisfy. ◀

For all T in $\mathcal{T}$, let $\mathcal{E}_m(T)$ denote the equivalence class of T for $\equiv_m$. We extend the map $\mathcal{E}_m$ to finite words, cyclic or not: if $w = w_1w_2 \ldots w_k$ is a word over $\mathcal{T}$, then $\mathcal{E}_m(w)$ is the word $\mathcal{E}_m(w_1)\mathcal{E}_m(w_2) \ldots \mathcal{E}_m(w_k)$. We further extend $\mathcal{E}_m$ to sets and multisets of words: if $Y = \{y_1, \ldots, y_n\}$ is a (multi)set of finite words over $\mathcal{T}$, then $\mathcal{E}_m(Y)$ denotes $\{\mathcal{E}_m(y_1), \ldots, \mathcal{E}_m(y_n)\}$. Since any graph may be viewed as a multiset of cyclic words over $\mathcal{T}$, it makes sense to write $\mathcal{E}_m(G)$ for all graph G.

▶ **Definition 5.2.2.** A *DULC* is a finite digraph that is a vertex-Disjoint Union of Labeled Cycles, where the labels are in $\mathcal{E}_m$.

All graphs of the form $\mathcal{E}_m(G)$ are DULC. Now define a new signature, with two binary relation synbols $=$ and $\to$ as before, and one unary relation symbol per element of $\mathcal{E}_m$. Formulas ϕ with this signature talk about graphs where vertices are $\mathcal{E}_m$-labeled (possibly with some multiply-labeled vertices, but this does not matter), such as DULC.

▶ **Theorem 5.2.3.** *For all m and all graphs G, G', if $\mathcal{E}_m(G) \equiv_m \mathcal{E}_m(G')$ then $G \equiv_m G'$.*

Proof. By Lemma 5.2.1, the set $\mathcal{E}_m$ is finite. Assume that $\mathcal{E}_m(G) \equiv_m \mathcal{E}_m(G')$; we show that $G \equiv_m G'$ with the Ehrenfeucht-Fraïssé method (see for instance [5, Theorem 2.2.8] or [11, Theorem 6.10]), by giving a winning strategy for Duplicator. Suppose that Spoiler plays somewhere in a tree t of G (the case of G' is symmetric). Let u be the node of $\mathcal{E}_m(G)$ corresponding to t. Imagine a game in $\mathcal{E}_m(G)/\mathcal{E}_m(G')$ where Spoiler just picked u in $\mathcal{E}_m(G)$ and let u' be the node picked by Duplicator in $\mathcal{E}_m(G')$ as a response (since $\mathcal{E}_m(G) \equiv_m \mathcal{E}_m(G')$, Duplicator has a winning strategy there). Let t' denote the tree of G' corresponding to u'

in $\mathcal{E}_m(G')$. Since u and u' have the same label (otherwise Duplicator would not win in the $\mathcal{E}_m(G)/\mathcal{E}_m(G')$ game), by definition of $\mathcal{E}_m$ we have $t \equiv_m t'$, so Duplicator has a winning strategy in the game t/t'. Therefore, in order to choose which node of t' to pick, Duplicator applies her t/t' winning strategy. The next turns go on similarly: Duplicator maintains a virtual game in $\mathcal{E}_m(G)/\mathcal{E}_m(G')$, and one more virtual game for each tree touched in the main game. In that manner, she can always retort to Spoiler in a way that maintains a local isomorphism. ◀

▶ **Theorem 5.2.4.** *For all integer m and all formula ψ of rank m, there is a formula $\mathcal{E}(\psi)$ such that for all graph G, we have $G \models \psi$ if and only if $\mathcal{E}_m(G) \models \mathcal{E}(\psi)$.*

Theorem 5.2.4 does not imply the converse of Theorem 5.2.3 because the rank of $\mathcal{E}(\psi)$ may be higher than m. We do not know whether the converse of Theorem 5.2.3 is true. To prove Theorem 5.2.4, we first rephrase Hanf's lemma for DULC.

▶ **Definition 5.2.5.** An *r-ball* in a graph, where r is an integer, is a subgraph induced by vertices linked to a given vertex by a path of length at most r. An *r-ball type* occuring in a graph is the graph-isomorphism class for a ball (for isomorphisms preserving the center).

▶ **Remark 5.2.6.** The possible 3^m-ball types in DULC are the pointed cycles of length at most $2 \cdot 3^m + 1$ and the path of length exactly $2 \cdot 3^m + 1$, pointed in its center.

▶ **Definition 5.2.7.** Let m be an integer, $e = 2 \cdot 3^m + 1$ the maximum number of vertices in a 3^m-ball of a DULC, and B_m the (finite) set of possible 3^m-balls types in DULC. Given a DULC H, its *profile* is the function $\pi_{H,m} : B_m \to \{0, \dots, m \cdot e\} \sqcup \{\omega\}$ defined as follows: $\pi_{H,m}(b)$ is the number of balls in H that are isomorphic to b in the case that it does not exceed $m \cdot e$, and ω otherwise.

We extend the usual order $\leq$ to $\{0, \dots, m \cdot e\} \sqcup \{\omega\}$ by making ω a global maximum. This yields a partial order over profiles: $\pi \leq \pi'$ if for all b, we have $\pi(b) \leq \pi'(b)$.

▶ **Lemma 5.2.8** (Hanf's lemma [10, Lemma 2.3] along with Remark 5.2.6)**.** *Let m be an integer, and H and H' be DULC. If $\pi_{H,m} = \pi_{H',m}$, then $H \equiv_m H'$.*

We call a profile *ϕ-positive* if its graphs are models of ϕ, and *ϕ-negative* otherwise (or simply *positive* and *negative* when no confusion ensues). We might write π_H for $\pi_{H,m}$ when m is clear from the context.

Proof of Theorem 5.2.4. Fix an integer m, and a formula ψ of quantifier rank m. Since there are finitely many possible DULC m-profiles, we can denote $\{\pi_0, \dots, \pi_{k-1}\}$, for some integer k, the set of profiles of $\mathcal{E}_m(G)$, where G ranges over graphs satisfying ψ. Now let $\mathcal{E}_m(\psi)$ be the formula expressing "this graph has profile either π_0, or π_1, …, or π_{k-1}."

The property "having profile π" is indeed expressible by a first-order formula: for all ball b, if $\pi(b) \neq \omega$ (respectively, $\pi(b) = \omega$), make a formula saying "there exist exactly $\pi(b)$ nodes (respectively, at least $m \cdot e + 1$ nodes) that are the center of a ball of type b." For a given ball type b, "being the center of a copy of b" is expressible as well: require that there exist $|b|$ distinct nodes, forming a cycle or a path (depending on b), with the right labels.

Now, if $G \models \psi$, by definition, $\{\pi_0, \dots, \pi_{k-1}\}$ contains the profile of $\mathcal{E}_m(G)$; thus $\mathcal{E}_m(G) \models \mathcal{E}(\psi)$. Conversely, if $\mathcal{E}_m(G) \models \mathcal{E}(\psi)$, then the profile of $\mathcal{E}_m(G)$ is the profile of some $\mathcal{E}_m(G')$, where $G' \models \psi$. By Lemma 5.2.8, $\mathcal{E}_m(G) \equiv_m \mathcal{E}_m(G')$, and by Theorem 5.2.3, $G \equiv_m G'$, so that $G \models \psi$. ◀

▶ **Proposition 5.2.9.** *If ϕ is an ω-nontrivial formula over DULC, then there is a nonempty DULC H and nonempty labeled cycles J' and D' such that either:*

(i) *for all $k \geq 0$ and $k' \geq 1$, we have $H \sqcup (\bigsqcup^k J') \models \phi$ and $H \sqcup (\bigsqcup^k J') \sqcup (\bigsqcup^{k'} D') \not\models \phi$; or*
(ii) *for all $k \geq 0$ and $k' \geq 1$, we have $H \sqcup (\bigsqcup^k J') \not\models \phi$ and $H \sqcup (\bigsqcup^k J') \sqcup (\bigsqcup^{k'} D') \models \phi$.*

Proof. Let m be the quantifier rank of ϕ. Since the profile of the disjoint union of two DULC is greater than either profile, there is a maximal DULC m-profile ρ. Assume that ρ is ϕ-negative (otherwise replace ϕ by $\neg\phi$). Since there are finitely many possible profiles and ϕ is ω-nontrivial, there is a positive profile having infinitely many models. Let π denote a maximal profile for this property.

If there is a cycle J' whose number of occurrences is unbounded among the models with profile π, then there is such a model H such that $\pi_{G'}(J') = \omega$, and $H \sqcup^k J'$ has the same profile as H for all k. If not, then the models with profile π have unbounded cycle lengths; so there is a model H and a word u over alphabet $\mathcal{E}_m$ of length $|\mathcal{E}_m|^e + 1$ such that u, as a path, occurs more than $m \cdot e$ times in H. For counting reasons, there is a word v of length e that occurs at least twice in u. So there is a cycle J' of length at least $e + 1$ whose label (as a word) is a factor of u. The graph $H \sqcup^k J'$ has the same profile as H for all k.

Observe that there is no profile greater than $\pi = \pi_H$ with finitely many models, so by construction, any profile greater than π is negative. Let D'' be a ball such that $\pi(D'') < \rho(D'')$ and D' any cycle containing D''. For all $k > 0$, we have $\pi_H < \pi_{H \sqcup^k D'}$, so the DULC $H \sqcup^k D'$ is a countermodel of ϕ. Since $\pi_{H \sqcup^k J'} = \pi_H$, we have $\pi_{H \sqcup^k J' \sqcup^{k'} D'} = \pi_{H \sqcup^{k'} D'}$ for all k, k'. ◀

5.3 Proof of Theorem 5.2

We proceed to a case disjunction. In a graph G, a *hanging trees* is a connected component of the graph obtained from G by removing all the edges in cycles. A *subtree* of a tree T is always *complete*, i.e., spanned by the set of nodes coaccessible from a given node (the root of the subtree). An *immediate subtree* is a tree whose root has depth 1 in the ambient tree.

Unbounded cycles

▶ **Proposition 5.3.1.** *Let ψ denote a formula such that ψ and $\neg\psi$ both have models with unbounded cycles. Then ψ-Dynamics is either* NP*-hard or* co-NP*-hard.*

Proof. Since both ψ and $\neg\psi$ have models with unbounded cycles, the projection $\phi = \mathcal{E}(\psi)$ is ω-nontrivial. Apply Proposition 5.2.9 to get a nonempty DULC H and nonempty cycles D' and J' with either the property (i) or (ii) from the proposition. Let m be the quantifier rank of ϕ, and G, D, J be nonempty graphs such that $\mathcal{E}_m(G) = H$, that $\mathcal{E}_m(D) = D'$ and that $\mathcal{E}_m(J) = J'$. By Theorem 5.2.4, G, J, D satisfy the corresponding property (i) or (ii) (with G, J, D, ψ instead of H, J', D', ϕ), because $\mathcal{E}_m$ behaves correctly with respect to $\sqcup$: $\mathcal{E}_m(G) \sqcup \mathcal{E}_m(G') = \mathcal{E}_m(G \sqcup G')$. The statement follows from Proposition 5.1.2 with $z = 1$. ◀

Unbounded degrees

By Lemma 5.2.1, the set $\mathcal{E}_m$ of equivalence classes of $\equiv_m$ for trees is finite. If T is a tree and $\alpha \in \mathcal{E}_m$, write $|T|_\alpha$ for the number of immediate subtrees of T of type α.

▶ **Lemma 5.3.2.** *Let T and T' be trees such that, for each $\alpha \in \mathcal{E}_m$, we have either $|T|_\alpha = |T'|_\alpha$ or $|T|_\alpha, |T'|_\alpha \geq m$. Then $T \equiv_m T'$.*

Proof. We give a winning strategy for Duplicator. If Spoiler plays in a subtree t of T that was never touched before (the case of T' is symmetric), then Duplicator chooses a subtree t' of T' such that $t \equiv_m t'$. By the Ehrenfeucht-Fraïssé theorem, Duplicator has a winning strategy for the game t/t', so she uses it to play her turn. If Spoiler plays subsequent turns in t or t', then Duplicator continues the game in t/t' with her winning strategy. Since the global game lasts m turns, by the condition on T and T', it is always possible for Duplicator to find a t' such that $t \equiv_m t'$ as needed. Thus this is indeed a winning strategy. ◀

▶ **Proposition 5.3.3.** *Let ψ denote a formula whose models have bounded cycles but unbounded degrees. Then ψ-Dynamics is* NP*-hard.*

Proof. Let ψ be a formula of quantifier rank m, whose models have unbounded degrees and bounded cycles, say of length at most ℓ.

By assumption, ψ admits a model with a hanging tree having a node v of degree at least $m \cdot |\mathcal{E}_m|$. Hence, the node v has at least m equivalent immediate subtrees $J_1 \equiv_m \cdots \equiv_m J_m$. Lemma 5.3.2 implies that, if we add more copies of J_1 as immediate subtrees of v in G, resulting in a graph G', then $G \equiv_m G'$. So, in particular, G' also satisfies ψ. Let J denote $\bigsqcup^{\ell+1} J_1$, with the pointed nodes of J being the roots of the copies of J_1. Let D be a cycle of length $|J|$, without pointed nodes. We have $|D| = |J| > \ell$. For all k, k', with v the pointed node of G, the graph $G \sqcup_2 (\bigsqcup_2^k J)$ is a model of ψ, while $G \sqcup_2 (\bigsqcup_2^k J) \sqcup_2 (\bigsqcup_2^{k'} D)$ is a countermodel by assumption on l. The statement follows from Proposition 5.1.2, with G, J, D as defined above and $z = 2$. ◀

Unbounded subtree depths

▶ **Lemma 5.3.4.** *Let ψ denote a formula whose models have bounded cycles, degrees, but unbounded hanging tree depths. Then there are a model G of ψ and two subtrees T, T' of a hanging tree of G such that $T' \subset T$ and $T \equiv_m T'$.*

Proof. Suppose that the models have bounded cycles. By Lemma 5.2.1, $\mathcal{E}_m$ is finite. For any graph G, call $E_m(G)$ the $\mathcal{E}_m$-labeled copy of G where each node v is labeled by the equivalence class of the subtree rooted in v – the ambient trees being the hanging ones. By assumption, the graphs $E_m(G)$, for $G \models \psi$, contain arbitrarily deep subtrees, whilst the number of colors in $\mathcal{E}_m$ is fixed and finite. By the pigeonhole principle, one of those subtrees in one of those models admits two nodes with the same label, the first one being an ancestor of the other one. The lemma then follows from the definition of the labels. ◀

▶ **Lemma 5.3.5.** *Let T be a tree, t a subtree of T and t' a tree such that $t \equiv_m t'$. If T' is the tree T where the occurences of t have been replaced with t', then $T \equiv_m T'$.*

The proof goes by induction on the depth of the root of t in T, and Lemma 5.3.2.

▶ **Proposition 5.3.6.** *Let ψ denote a formula whose models have bounded degrees, but unbounded hanging tree depths. Then ψ-Dynamics is* NP*-hard.*

Proof. Let ψ be a formula of quantifier rank m, whose models have unbounded hanging tree depths, and bounded degrees, say by d.

By Lemma 5.3.4, there is a model $\widetilde{G}$ of ψ that contains a tree T (i.e. a subtree of a hanging tree), which in turn has a subtree T', such that $T' \equiv_m T$, and such that the only vertices of T and T' linked to the rest of the graph are their roots. Call T'' the tree $T \setminus T'$. Let G denote $(\widetilde{G} \setminus T') \sqcup T'$: it is a disconnected graph. We equip it with two pointed nodes, u and v, like in Definition 5.1.1 (case $\sqcup_3$). The pointed node u of G is the leaf of T that should have been the parent of the root of T'. The pointed node v of G is the root of the disconnected copy of T'. See Figure 3 for an illustration.

Now let $T''_0 = T''$, and T''_{n+1} denote the tree T where T' have been replaced by a copy of T''_n. For all n, equip the graph T''_n with two pointed nodes: the node u is the leaf where another copy of T'' would be inserted to build T''_{n+1}; the node v is the root. See again Figure 3 for an illustration.

Let J be the graph T''_{d+2}, so that $|J| > d+1$. By Lemma 5.3.5, we have $G \sqcup_3 (\bigsqcup_3^k J) \equiv_m \widetilde{G}$ for all integer k. Let D be a tree of depth 1 having $|J|$ nodes, i.e., it consists only of a root and its direct children; its pointed node u is any leaf, and its pointed node v is the root. The tree D has degree at least $d+1$. Therefore, for all k, k', the graph $G \sqcup_3 (\bigsqcup_3^k J)$ is a model of ψ, while the graph $G \sqcup_3 (\bigsqcup_3^k J) \sqcup_3 (\bigsqcup_3^{k'} D)$ is a countermodel. The statement follows by Proposition 5.1.2 with $z = 3$. ◀

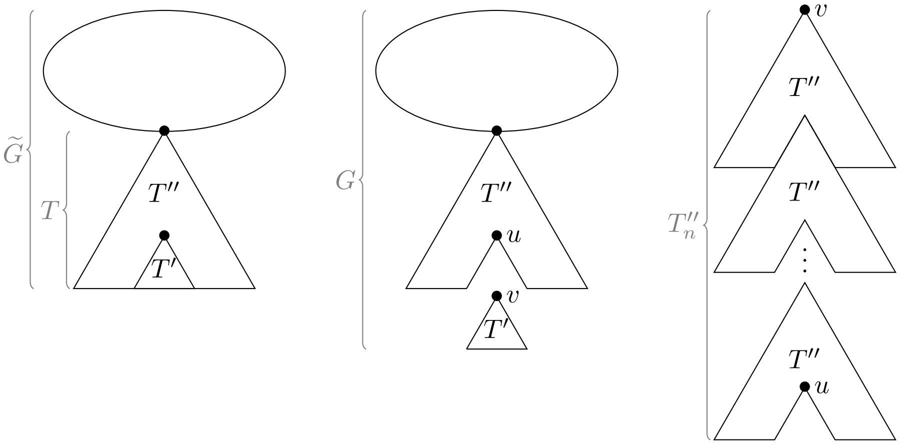

Figure 3 Illustration of the construction in the proof of Proposition 5.3.6. The node v of each graph transitions to the node u of another one.

Unbounded number of occurrences of each connected component

Here, *connected* means *strongly connected.* The *number of occurrences* of a connected component C in a graph G is the number of connected components of G isomorphic to C.

▶ **Lemma 5.3.7.** *Let G and J be graphs and m an integer. For all integers $k, k' \geq m$, we have $G \sqcup (\bigsqcup^k J) \equiv_m G \sqcup (\bigsqcup^{k'} J)$.*

Proof of Lemma 5.3.7. We give a winning strategy for Duplicator. If Spoiler plays in either copy of G, then Duplicator picks the same node in the other copy of G. If Spoiler plays in a copy of J that was never touched before, then Duplicator chooses a fresh copy of J in the other graph and picks the same node there. If Spoiler plays in a copy of J that was already touched before, then Duplicator chooses the same copy of J as in the previous moves and picks the same node there. Since there are at least m copies of J on both graphs and only m turns in the game, this is indeed a winning strategy. ◀

▶ **Proposition 5.3.8.** *Let ψ is a formula whose models have bounded cycles, but unbounded number of occurrences of each connected component. Then ψ-Dynamics is* NP*-hard.*

Proof. Let ψ be a formula of quantifier rank m, whose models have unbounded number of occurrences of each connected component, and bounded cycles, say of length at most ℓ. By our assumptions on ψ, there are graphs G and J' such that $G \sqcup (\bigsqcup^m J')$ is a model. Let J

denote $\bigsqcup^{\max(\ell+1,m)} J'$, and D a cycle of length $|J|$. For all k, k', by Lemma 5.3.7, the graph $G \sqcup (\bigsqcup^k J) = G \sqcup (\bigsqcup^{k \cdot \max(\ell+1,m)} J')$ is a model of ψ. On the other hand, by assumption on $\ell < |J|$, the graph $G \sqcup (\bigsqcup^k J) \sqcup (\bigsqcup^{k'} D)$ is a countermodel.

The statement follows from Proposition 5.1.2 with G, J, D defined above and $z = 1$. ◀

Combining the cases

▶ **Lemma 5.3.9.** *Every formula with infinitely many models has models with either unbounded cycles, unbounded degrees, unbounded hanging tree depths, or an unbounded number of occurrences of each connected component.*

Proof. The number of nonisomorphic connected graphs with a cycle of length at most ℓ, degree at most d and hanging tree depth at most h is bounded by ł $\cdot\, d^{h+1}$. Thus there are only finitely many graphs with bounded cycles, degrees, subtree depths and number of occurrences of connected components. ◀

Lemma 5.3.9 concludes the proof of Theorem 5.2: if the formula and its negation both have unbounded cycles, then Proposition 5.3.1 applies; otherwise one among Propositions 5.3.3, 5.3.6 and 5.3.8 applies to either the formula or its negation.

▶ **Remark 5.3.10.** The machinery developed to prove Theorem 5.2 is rather flexible.

In particular, it remains true if restricted to deterministic automata networks, and also to the limit subgraphs ($\mathcal{G}_f^\omega$) instead of transition digraphs ($\mathcal{G}_f$). However, the meaning of "ω-nontrivial" changes: it respectively means "having infinitely many bijective (counter)models" and "having infinitely many networks whose limit graph is a (counter)model."

Both transition digraphs of bijective networks and limit graphs are merely disjoint unions of unlabeled cycles. Thus, Proposition 5.2.9 and Proposition 5.1.2 may be reused and the proof is similar.

6 First-order dynamical properties are arbitrarily high in PH

The previous section gave a lower bound for the ψ-Dynamics problem. Here, we give tighter bounds. As a consequence of those bounds, the AN-Dynamics problem, which is similar to ψ-Dynamics except that ψ *is part of the input,* is hard.

▶ **Theorem 6.1.** *For all even integer N, there is a formula ψ_N such that ψ_N-Dynamics Σ_{N+1}-complete.*

▶ **Theorem 6.2.** *The following problem is* PSPACE*-complete:*

AN-Dynamics
Input: an automata network f and a first-order formula ψ.
Question: does $\mathcal{G}_f \models \psi$?

The proofs rely on the following constructions. Let $N \geq 1$, and S be a QBF formula of the form $\exists b_1, \forall b_2, \ldots, \exists b_{N+1} R(b_1, \ldots, b_{N+1})$. Call C the set $\{\top, \bot\} \sqcup \bigsqcup_{i=1}^{N+1} \{0,1\}^i$, where $\top, \bot$ are fresh symbols. Observe that $|C| = 2^{N+2}$ and define f_S the ANU with $N+2$ nodes over alphabet $\{0,1\}$ that realizes the function $f_S : C \to C$ defined as follows. For arbitrary bits $b_1, \ldots, b_{N+1}$, set $f_S(\bot) = \bot$, $f_S(\top) = \top$, $f_S((b_1)) = \top$, and:

$$f_S((b_1, \ldots, b_i)) = (b_1, \ldots, b_{i-1}) \quad f_S((b_1, \ldots, b_{N+1})) = \begin{cases} (b_1, \ldots, b_N) \text{ if } R(b_1, \ldots, b_{N+1}) \\ \bot \text{ otherwise.} \end{cases}$$

Intuitively, the dynamics of f_S consists of two upward trees: one rooted in $\top$, of depth $N+1$, whose leaves are the Boolean tuples $(b_1, \dots, b_{N+1})$ that satisfy R; and one rooted in $\bot$, of depth 1, whose leaves are the Boolean tuples $(b_1, \dots, b_{N+1})$ that falsify R. The only part of f_S that depend on S merely evaluates R, so circuits encoding f_S can be produced in polynomial time given S.

Now define the formula ψ_N as follows (observe that ψ_N depends only on N):

$$\begin{aligned}\psi_N =& \exists x_0, x_1, x'_2 : x_0 \neq x_1 \wedge x'_2 \to x_1 \to x_0 \to x_0 \\ & \wedge \forall x_2 \to x_1 : \exists x_3 \to x_2 : \dots \forall x_N \to x_{N-1} : \exists x_{N+1} \to x_N : \mathsf{true},\end{aligned}$$

where "$\exists x \to y : \phi$" and "$\forall x \to y : \phi$" stand for "$\exists x : (x \to y) \wedge \phi$"and "$\forall x : (x \to y) \implies \phi$"; and where "$x \to y \to z$" stands for "$x \to y \wedge y \to z$". Observe that ψ_N is a Σ_{N+1}-formula. Besides, when evaluating ψ_N in $\mathcal{G}_{f_S}$, the first line ensures that x_0 is a fixed point with an ingoing path of length 2, so it has to be $\top$. The rest of the formula straightforwardly implements S, by linking Booleans into a configuration $(b_1, \dots, b_{N+1})$ where the "$x_{N+1} \to x_N$" part ensures that $R(b_1, \dots, b_{N+1})$, by definition of f_S. Hence we have the following lemma, that implies both Theorem 6.1 and 6.2.

▶ **Lemma 6.3.** **(a)** *The network f_S satisfies $\mathcal{G}_f \models \psi_N$ if and only if S is a true QBF.*
(b) *Given a* QBF(Σ_{N+1}) *formula S, the network f_S can be produced in polynomial time.*

7 Conclusion

Our goal was to obtain broad complexity lower bounds for dynamical properties of automata networks. However, as explained in the introduction, there is a large degree of freedom in the formalization of Metatheorem 1.2. We do not claim that the results above are the only Rice-like theorems on automata networks worth investigating.

It would be interesting to know how various restrictions on the AN may lessen the complexity of those problems. For instance, if we restrict ourselves to AN whose interaction graph has bounded degree, then the question "does this AN compute a constant function?" becomes testable in polynomial time, while it is first-order expressible and nontrivial.

Another restriction pertains to the set of states of the nodes. If we restrict ourselves to ANU, i.e., automata networks where all nodes have the same alphabet $Q = \{0, \dots, q-1\}$ for some positive integer q, then the concept of "ω-nontriviality" changes. Indeed, some first-order formulas have infinitely many models and countermodels, but no model with uniform alphabet. The proof of Lemma 5.1.4 does not seem to generalize easily to that case, because finding x and δ becomes an open challenge.

References

1 J. Aracena. Maximum number of fixed points in regulatory Boolean networks. *Bull. Math. Biol.*, 70:1398–1409, 2008.

2 B. Borchert and F. Stephan. Looking for an analogue of Rice's theorem in circuit complexity theory. *Mathematical Logic Quarterly*, 46(4):489–504, 2000. doi:10.1002/1521-3870(200010)46:4<489::AID-MALQ489>3.0.CO;2-F.

3 P. Cull. Linear analysis of switching nets. *Biol. Cybernet.*, 8:31–39, 1971.

4 J. Demongeot, M. Noual, and S. Sené. Combinatorics of Boolean automata circuits dynamics. *Discr. Appl. Math.*, 160:398–415, 2012.

5 H.-D. Ebbinghaus and J. Flüm. *Finite Model Theory.* Springer-Verlag, 2nd edition, 1995. doi:10.1007/3-540-28788-4.

6 B. Elspas. The theory of autonomous linear sequential networks. *IRE Trans. Circ. Theory*, 6:45–60, 1959.

7 C. Espinosa-Soto, P. Padilla-Longoria, and E. R. Alvarez-Buylla. A gene regulatory network model for cell-fate determination during *Arabidopsis thaliana* flower development that is robust and recovers experimental gene expression profiles. *The Plant Cell*, 16:2923–2939, 2004.

8 E. Goles and S. Martinez. *Neural and Automata Networks: Dynamical Behavior and Applications*. Kluwer Academic Publishers, 1990.

9 S. W. Golomb. *Shift Register Sequences*. Holden-Day Inc., 1967.

10 W. Hanf. Model-theoretic methods in the study of elementary logic. In J.W. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models*, pages 132–145. North-Holland, 1963. doi:10.1016/B978-0-7204-2233-7.50020-4.

11 N. Immerman. *Descriptive Complexity*. Springer-Verlag, 1999. doi:10.1007/978-1-4612-0539-5.

12 J. Kari. Rice's theorem for the limit sets of cellular automata. *Theoretical Computer Science*, 127:229–254, 1994.

13 G. Karlebach and R. Shamir. Modelling and analysis of gene regulatory networks. *Nature Rev. Mol. Cell Biol.*, 9:770–780, 2008.

14 S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969. doi:10.1016/0022-5193(69)90015-0.

15 S. C. Kleene. *Automata Studies*, chapter Representation of events in nerve nets and finite automata, pages 3–41. Princeton University Press, 1956.

16 W. S. McCulloch and W. H. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5:115–133, 1943.

17 L. Mendoza and E. R. Alvarez-Buylla. Dynamics of the genetic regulatory network for *Arabidopsis thaliana* flower morphogenesis. *J. Theoret. Biol.*, 193:307–319, 1998.

18 H. G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74:358–366, 1953. doi:10.1090/S0002-9947-1953-0053041-6.

19 A. Richard. Local negative circuits and fixed points in non-expansive Boolean networks. *Discr. Appl. Math.*, 159:1085–1093, 2011.

20 F. Robert. *Discrete Iterations: A Metric Study*. Springer Verlag, 1986.

21 J. H. Silverman. *A friendly introduction to number theory*. Pearson Education, 4th edition, 2012.

22 L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time(preliminary report). In *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing*, STOC '73, pages 1–9, New York, NY, USA, 1973. ACM. doi:10.1145/800125.804029.

23 D. Thieffry and R. Thomas. Dynamical behaviour of biological regulatory networks – II. Immunity control in bacteriophage lambda. *Bull. Math. Biol.*, 57:277–297, 1995.

24 R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42:563–585, 1973. doi:10.1016/0022-5193(73)90247-6.