Internship Report: First-Order questions on the dynamics of automata networks with distinguished configurations

Aliénor Goubault–Larrecq

Supervised by:
Kévin Perrot
Enrico Porreca

LIS

from 21/02/2022 to 20/07/2022

Abstract

We prove general complexity lower bounds on automata networks, in the style of Rice's theorem, as an expansion of a result by Gamard, Guillon, Perrot and Theyssier. Their result was about first-order formula on the dynamics of automata networks, but it was up-to isomorph. A question left open was how to differentiate isomorphic graphs.

Our main result is that testing any fixed graph first-order formula on the dynamics of a deterministic automata network is either trivial or P-complete, if we distinguish a finite number of configurations, especially if we can recognize the minimal configuration. We also give some properties on first-order formula with respect to the comparison of configurations, such that it can give some avenues of reflexion about the extension of our problem to formulas with an order relation.

Contents

1 Introduction **1**

2 Definitions **2**

2.1 Automata networks . . . 2

2.2 Graph First-Order Logic . . . 3

2.3 Complexity classes . . . 4

2.4 Nomenclature used for logic circuits . . . 5

3 State of the art **6**

4 FO questions on signature $\{=, \rightarrow, \trianglelefteq\}$ **7**

4.1 Some FO questions on signature $\{=, \rightarrow, \trianglelefteq\}$ are P-complete . . . 7

4.2 FO questions on signature $\{=, \trianglelefteq\}$ are ω-trivial . . . 8

4.2.1 FO questions on signature $\{=, \leq_b\}$ are ω-trivial . . . 8

4.2.2 FO questions on signature $\{=, \leq_t\}$ are ω-trivial . . . 9

4.3 Properties . . . 9

4.3.1 Find a model and a counter-model with exactly one difference . . . 9

4.3.2 Permutations and Hamming weight . . . 10

5 FO questions on signature $\{=, \rightarrow, \mathcal{Z}\}$ are P-hard **11**

5.1 General proof . . . 11

5.2 Extention . . . 19

6 Conclusion **20**

A Last cases of the proof about FO questions on signature $\{=, \rightarrow, \mathcal{Z}\}$ **22**

B Circuit for the reduction of signature $\{=, \rightarrow, \mathcal{Z}\}$ **24**

C FO questions on signature $\{=, \rightarrow\}$ are P-hard **24**

1 Introduction

An *automata network* is a directed graph $G = (V, E)$, together with a finite set of so-called *states* S_v for each vertex $v \in V$ that is representing the states of an *automaton*, and a function that expresses how the state at each automaton v evolves as a function of the states of its predecessors. The tuple of states of the automaton of an automata network at a time t is called a configuration. An automat network can be deterministic or not. Also, in the cases where each automaton has two states, an automata network is called a *Boolean automata network*.

Initially, Boolean automata networks were introduced by McCulloch and Pitts in 1943 as a formal model of another type of networks: the neural networks [8]. Later, general automata networks were introduced in theoretical biology, in order to study dynamics of gene expression, their activations and inhibitions, as formalized by Kauffman in 1969 [6] and Thomas in 1973 [13].

Since then problems on automata networks have been studied with the general aims of being applicable to biology, while having theoretical results on its structure. A lot of studied problems about automata networks are about their *dynamics*, which is the function that computes the next states of every automata at each step. In particular, given a property, some people asked what the complexity to determine whether an automata network verifies this property is. For example, Alon proved in 1985 that knowing whether an automata network admits a fixed point (*i.e.* a stable configuration) is complete for the complexity class NP, and is thus as hard to solve as all the problems of this class [1]. Other problems on fixed points have been studied such as counting the number of fixed points [9], or computing *limit set* (the set of configurations visited infinitely many times if they are visited once) [4], and about the *basin of attraction* of a configuration x (set of configurations that can reach x) [3].

Gamard, Guillon, Perrot and Theyssier made a generalisation in 2021 by studying the complexity of the problem, that given a property expressed with graph first-order logic, whether a graph verifies this property or not [4]. They showed that for properties on the dynamics of graphs, in first-order logic, it is always trivial (solvable in constant time) or at least as hard as problems in the classes NP or coNP, which is a Rice-like theorem (in the spirit of Rice's theorem in computability theory [12]).

They noticed that first order graph question on their signature are invariant under isomorphim, in other words, it is not possible to differenciate configurations. This is an issue for biology-applications, since the labels of the configurations, for instance the gene activation states, can carry an information that is not interchangeable. That is why distinguishing configurations of these networks is interesting, in order to distinguish genes. A goal of this report is to show what relations it is possible to add to the logic's signature in order to differenciate some configurations.

Outline. We first set out all the preliminary notions in Section 2., in particular concerning automata networks, graph first-order logic and complexity theory. Then, we present the state of the art which led to our problem in Section 3. We start describing our own controbutions, and we show some properties of questions when we add an order on the configurations in Section 4. We prove a general complexity lower bound for the problem when we have a unary relation to distinguish the configuration where all the automata are in the state $\mathtt{0}$, and give an extension when we distinguish a finite number of configurations in Section 5. We also give in Appendix C an improvement of the proof of the Rice-like theorem by Gamard, Guillon, Perrot and Theyssier, with a reduction of better complexity.

2 Definitions

For two natural numbers $n, q \in \mathbb{N}$, we will use the following notations to define sets of integers: $[n] = \{1, \ldots, n\}$ and $[\![q]\!] = \{0, \ldots, q-1\}$. We write $A_i = [\![q_i]\!]$ with $q_i \in \mathbb{N}$ for all $i \in [n]$. Given a set X, we write $|X|$ the cardinality of X, *i.e.* the number of elements in X.

A graph G is defined by two sets: the set of vertices $V(G)$ and the set of edges or arcs depending whether G is directed or not $E(G)$. To define a graph we write $G = (V(G), E(G))$. If G is directed, $u, v \in V(G)$, and there is an arc from u to v we write that $(u, v) \in E(G)$. If G is not directed and there is an edge between u and v, we write that $\{u, v\} \in E(G)$. Two graphs G and H are *isomorphic* if H can be obtained after renaming all the vertices of G, *i.e.* there is a bijection $\theta : V(G) \to V(H)$ such that $(u, v) \in E(G) \iff (\theta(u), \theta(v)) \in E(H)$ (and similarly in the undirected case). Concerning the size of a graph we consider that $|G| = |V(G)|$ in this report.

2.1 Automata networks

A *deterministic automata network* (abbreviated by AN here) of size n is a function $f : X \to X$ where $X = \prod_{i \in [n]} A_i$ is the set of the *configurations* of the system, and then A_i is the set of states of the i^{th} automaton of the network. In the Boolean case, each automaton can only have two states $A_i = \{0, 1\}$, hence $X = \{\mathtt{0}, \mathtt{1}\}^n$.

By default, we will consider in this report that the automata networks have one automaton where $n = 1$ (cf. Remark 1).

The funtion f can be split into a family of *local functions* $\{f_i\}_{i \in [n]}$, where $\forall i \in [n], f_i : A \to A_i$. Hence f_i returns the state of the i^{th} automaton at the next step. We can also retrieve f from all the local functions since $\forall x \in X, f(x) = (f_1(x), f_2(x), ..., f_n(x))$. We remark that f_i does not necessarily depend on the previous state of all the other n automata.

The function f can be represented by a graph G_f called *interaction digraph*, such that $G_f = ([n], \mathcal{I})$, where $\mathcal{I}$ is the set of pairs (i, j) such that, for some $a, b \in X$ with $a_k = b_k$ for every $k \neq i$, we have $f_j(a) \neq f_j(b)$. We can also represent the graph of the configurations $\mathcal{G}_f$ of an AN f, such that $V(\mathcal{G}_f) = X$ and $E(\mathcal{G}_f) = \{(x, f(x)), \forall x \in V(\mathcal{G}_f)\}$. We call it *dynamics* or *transition digraph.*

An automata network is encoded as a tuple of n Boolean circuits, one for each of its local functions, with a total size not greater than $n2^n$, *i.e.* the size of the dynamics $\mathcal{G}_f$, *i.e.* the logarithm of the number of functions on $\{0,1\}^n \to \{0,1\}^n$, since the truth table of each local function have a size of at most 2^{2^n}.

Remark 1. *We can transform any AN f into an equivalent network g with only one automaton. Let X be the set of configurations of f. There exists an AN g with one automaton with the set of states $[\![|X|]\!]$ with dynamics $\mathcal{G}_f = \theta(\mathcal{G}_g)$, where θ is a renaming of the vertices according to the trivial bijection between X and $[\![|X|]\!]$.*

Hence we will mainly consider automata networks on one automaton, whose state is encoded in binary. By convention, configurations x such that $x \geq |X|$ does not have a meaning for this automata network (with $X = A_1$ in the case of one automaton).

We note that Boolean automata networks are exactly the automata networks whose number of configurations is a power of 2.

2.2 Graph First-Order Logic

If P is a property that automata networks may or may not satisfy, and f is an automata network, then we write $f \vDash P$ if f satisfies P, and $f \nvDash P$ otherwise. We say that f is a *model* of P in the first case, and a *counter-model* otherwise. This is an abuse of notation, and we need to know the exact nature of P to know its precise meaning. In particular, we will study properties expressible on *Graph First-Order Logic* over a signature $\mathcal{S}$.

A *signature* $\mathcal{S}$ gives symbols to the formulas we can construct. In particular it describes the possible *atomic relations* on configurations (also called *atoms*). In this report, we will mainly add three symbols in the signature, depending on the AN given in input f:

- $=$ is a binary relation of the equality between configurations;
- $\to$ is a binary relation such that $x \to y$ if and only if $y = f(x)$ (or $(x,y) \in E(\mathcal{G}_f)$);
- $\trianglelefteq$ is a binary relation of order between configurations. We use it to talk indifferently of the bitwise partial order of binary strings denoted by $\leq_b$, or the total order of integers denoted by $\leq_t$.

To say that we consider three relations $=, \to, \trianglelefteq$ in a signature we write the signature $\mathcal{S} = \{=, \to, \trianglelefteq\}$.

First-Order formulas, abbreviated by FO, are formulas that are expressible with the existential quantification $\exists$ on vertices (configurations); conjonction $\wedge$, and negation $\neg$. We use syntactical shortcuts to express the universal quantification $\forall$, the disjunction $\vee$, the implication $\implies$, and the equivalence $\iff$; that are all derived from the first three symbols.

The *quantifier rank* of a formula ψ is its depth of quantifier nesting, see for example [7, Definition 3.8]. If G and G' are two structures, we write $G \equiv_m G'$ if and only if they satisfy the same formulas of quantifier rank m, we say that they are *partially isomorphic*. We write $\psi \equiv \varphi$ if the two formulae ψ and φ are equivalent, *i.e.* if they have the same model set.

2.3 Complexity classes

Complexity classes are classes of problems characterized by their complexity, in terms of resources needed to be solved with a Turing machine: the ressources which we analyse here, are time and memory (space).

Some classes are defined with *oracles* that are seen as black boxes a Turing machine can use to solve certain problems in a single operation: we note X^Y for the class of problems in X using an oracle to do an operation in Y.

We define the complexity classes that we will need in this report:

- L: class of problems solvable in logarithmic space on a deterministic Turing machine; *i.e.* on an input of size n, it can be solved sequentially in space $O(\log n)$;
- NC: class of problems solvable in polylogarithmic time on a parallel computer with a polynomial number of processors; *i.e.* there exist two constants k, c such that on an input of size n it can be solved in time $O((\log n)^k)$ using $O(n^c)$ processors.
- P: class of problems solvable by a deterministic Turing machine in polynomial time; *i.e.* there exists a constant k such that on an input of size n it can be solved sequentially in time $O(n^k)$;
- NP: class of problems solvable by a non-deterministic Turing machine in polynomial time; *i.e.* there exists a constant k such that on an input of size n it can be solved sequentially and non-deterministically in time $O(n^k)$;
- coNP: class of problems whose complement is in NP;
- Σ_i^{P} with $i \in \mathbb{N}$ is the i^{th} level of the *polynomial hierarchy*, defined inductively by $\Sigma_0^{\mathsf{P}} = \mathsf{P}$ and $\Sigma_{i+1}^{\mathsf{P}} = \mathsf{NP}^{\Sigma_i^{\mathsf{P}}}$;
- Π_i^{P}, complementary of Σ_i^{P}, defined inductively by $\Pi_0^{\mathsf{P}} = \mathsf{P}$ and $\Pi_{i+1}^{\mathsf{P}} = \mathsf{coNP}^{\Pi_i^{\mathsf{P}}}$.

To prove that a problem belongs to a class, we often use a *reduction.* A reduction is a transformation of a problem X to another problem Y. If X can be reduced to Y then it means that X is not more difficult to solve than Y. There are different methods of reduction, but in this report we will focus on the *many-one reductions.* If the reduction is done with respect to a class $\mathbf{C}$, then there is an algorithm, with complexity being similar to problems of $\mathbf{C}$, that transforms the inputs to problem X into inputs to problem Y, such that the two problems with these inputs have the same output. If X reduces to Y with respect to $\mathbf{C}$, we write it $X \leq^{\mathbf{C}} Y$.

A problem X is said to be *hard* for a complexity class $\mathbf{C}$ if every problem in $\mathbf{C}$ can be reduced to X. This means that no problem in $\mathbf{C}$ is harder than X, since an algorithm for X allows us to solve any problem in $\mathbf{C}$. The reduction needs to be adapted depending on the class $\mathbf{C}$, for complexity classes larger than P, we often use polynomial-time reductions, in particular for NP problems. To show that a problem is P-hard, we usually use NC or L reductions. A problem X is said to be *complete* for a complexity class $\mathbf{C}$ if X is in $\mathbf{C}$ and is hard for $\mathbf{C}$.

To reduce a problem to a certain class, we often use problems of reference. In particular, for P-reduction (to prove that a problem is P-hard), we often use **Circuit-Value-Problem**. We use the same definition of *Boolean circuit* as defined by [7, Section 6.2],on n inputs and with the logical relations $\wedge, \vee, \neg$. For a circuit C on n inputs and x a vector on n bits, we call $C(x) \in \{\mathtt{1}, \mathtt{0}\}$ the output of the circuit C on the input x.

Circuit-Value-Problem (CVP)
Input: Boolean circuit C on n inputs and x a vector on n bits.
Question: $C(x) = \mathtt{1}$?

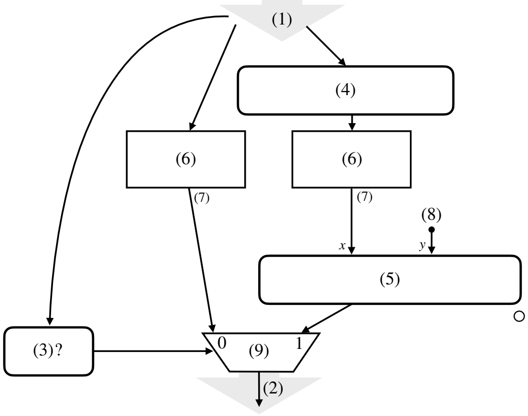
Theorem 1 ([10, Theorem 8.1])**. CVP** *is* P*-complete with respect to* L*.*

Theorem 2. CVP *with fixed input* $x = \mathtt{0}^n$ *is* P*-complete.*

2.4 Nomenclature used for logic circuits

We will represent some logic circuit in this report. Logic circuits are made to be read from top to bottom. The arrows represent the information flow direction: if an arrow goes to a box, it will be considered as an input of the function in the box, while if it goes out, it will be considered as an output.

We give an explicative layout in Figure 1.



- (1) represents the input of the circuit on the top with a grey arrow;
- (2) at the bottom, is the output of the circuit;
- (3), (4) and (5) are defined by a function with variables denoted by x, y or z. There are different types of functions: the ones such as (3) whose only outputs can be $\mathtt{0}$ or $\mathtt{1}$; the ones such as (4) that are defined by a function under the form $x \mapsto f(x)$, with x an input; (5) is similar to (4) but the inputs are divided, and is under the form $x, y \mapsto f(x, y)$, in this case we specify which input corresponds to which variable;
- (6) are pre-existing circuits that we copy (it appears two times here but it is not necessarily for the same circuit);
- (7) is used to recall an output in the circuit;
- (8) is when a constant is used as an input;
- (9) is a 2 to 1 multiplexer: *i.e.* from two inputs, it will output one of them depending on the bit given from the left or the right.

Figure 1: Explicative circuit.

Most of the time, we don't specify in each arrow how many bits it bears; we construct circuits such as the the output of a box has as many bits as the input of the box an arrow leads to. If this is not the case, we consider that either the vector born by the arrow will be truncated, or padded with $\mathtt{0}$s to have the correct number of bits.

3 State of the art

Several problems on the dynamics of automata networks have been studied, we give here some examples of problems about properties expressible in graph first-order logic.

In the dynamics graph $\mathcal{G}_f$, a *limit cycle* is a cycle of the graph. The set of the limit cycles of size k is written $\mathcal{C}_f^k$. We note that $\mathcal{C}_f^1$ is the set of fixed points in $\mathcal{G}_f$. We can questions about the existence of cycles of size k (k being fixed in the problem):

Limit Cycle of size k (k-CL)
Input: a Boolean AN f (local functions encoded as circuits).
Question: does $|\mathcal{C}_f^k| > 0$?

Alon studied this problem for $k = 1$, about fixed points and showed the following result in 1985:

Theorem 3 ([1])**.** 1**-CL** *is* NP*-complete, even with the promise that the maximal degree of* G_f *is inferior or equal to 2.*

Later, Bridoux, Gaze-Maillot, Perrot and Sené studied the problem for all k:

Theorem 4 ([2])**.** k**-CL** *est* NP*-complete* $\forall k \in \mathbb{N}^+$*, even with the promise that the maximal degree of* G_f *is inferior or equal to 2.*

Other problems expressible in graph first-order logic can be in coNP, such as the bijectivity. We can verify that an AN is bijective by only verifying the injectivity expressed with $\psi \equiv \forall x, x', \exists y, y', (x \neq x' \wedge x \to y \wedge x' \to y') \implies y \neq y'$, since:

Theorem 5 ([11])**.** *An AN is bijective if and only if it is injective.*

Bijectivity
Input: a Boolean AN f (local functions encoded as circuits).
Question: Is f bijective?

Theorem 6 ([11])**. Bijectivity** *is* coNP*-complete.*

In 2021, Gamard, Guillon, Perrot and Theyssier wrote an article about the complexity of closed first-order logic formulas over the signature $\{=, \to\}$ (both binary relations) of transition digraphs, which is a generalization of the previous problems, and prove a Rice-like theorem about the following problem:

ψ-dynamics
Input: an automata network f (local functions encoded as circuits).
Question: does $\mathcal{G}_f \vDash \psi$?

Definition 1. *A formula ψ is ω-nontrivial if there are infinitely many models and infinitely many countermodels.*

Theorem 7 ([4])**.** *If ψ is ω-nontrivial, then* ψ**-dynamics** *is either* NP*- or* coNP*-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 $O(1)$.

They noticed that ψ formula on FO logic over the signature $\{=, \to\}$ is up to isomorphism, ie. for a graph $\mathcal{G}_g$ which is the exact same graph $\mathcal{G}_f$ with the vertices renamed we have $\mathcal{G}_g \vDash \psi \iff \mathcal{G}_f \vDash \psi$.

They proved a complexity's lower bound, and they also said an evident upper bound is that it is always in $\mathsf{PH} = \cup_{i \in \mathbb{N}} \Sigma_i^{\mathsf{P}}$ (all the problems are in the polynomial hierarchy), and proved that both bounds were optimal:

Theorem 8 ([4, 11]). *For all $N \in \mathbb{N}^*$, there is a formula ψ_N such that ψ_N-Dynamics is $\Sigma_{N+1}^{\mathsf{P}}$-complete.*

4 FO questions on signature $\{=, \to, \trianglelefteq\}$

We recall that we write $\trianglelefteq$ to talk indifferently of a partial order or a total order, where the bitwise partial order of binary strings is denoted by $\leq_b$, and the total order of integers is denoted by $\leq_t$..

4.1 Some FO questions on signature $\{=, \to, \trianglelefteq\}$ are P-complete

Theorem 9. *For all of the following ψ, the problem ψ-**dynamics** is P-complete.*

- $\forall x : (\forall y : x \trianglelefteq y) \implies x \to x$, i.e. *configuration $\mathtt{0}^n$ is a fixed point;*
- $\forall x : \forall y : (x \leq_t y \land (\forall z : x \leq_t z \land z \leq_t y \implies z = x \lor z = y)) \implies (\exists y_1 : \exists y_2 : y \to y_1 \land y_1 \to y_2 \land y_2 \to y \land y_1 \neq y_2 \land y_1 \neq y \land y_2 \neq y)$, i.e. *configuration $\mathtt{1}$ (in the total order, $\mathtt{1}$ is the smallest configuration after $\mathtt{0}$) belongs to a cycle of size 3.*

Proof.

- For $\forall x : (\forall y : x \trianglelefteq y) \implies x \to x$, first, given the graph $\mathcal{G}_f$ in input, it is immediately in P, since we just need to compute $f(\mathtt{0})$, which can be computed in at most polynomial time.

  Let C be an instance of **Circuit-Value-Problem** with n bits of inputs, with fixed input $\mathtt{0}^n$. We will construct a Boolean automata network of size n, such that for every local function $f_i, i \in [n]$ (cf. the circuit in the left of Figure 2):

  $$f_i(x) = \neg C(\mathtt{0}^n).$$

  It is immediate by construction that $\mathcal{G}_f \vDash \psi$ if and only if $C(\mathtt{0}^n) = 1$.

  The reduction is in L: the circuit of the automata network can be constructed in $O(\log(|C|))$ space, where $|C|$ is approximately the number of gates in C.
- First it is in P since, given the graph $\mathcal{G}_f$ in input, we just need to compute $f(\mathtt{1}), f(f(\mathtt{1}))$ and $f(f(f(\mathtt{1})))$, which can be done in at most polynomial time. We use the same method as the previous point, except that if we write each configuration $x = x_n, ..., x_2, x_1$, we create a cycle between each configurations who ends by $\mathtt{10}$, $\mathtt{01}$, $\mathtt{11}$ such that $\mathtt{01} \to \mathtt{10} \to \mathtt{11} \to \mathtt{01}$ without changing the rest. Hence

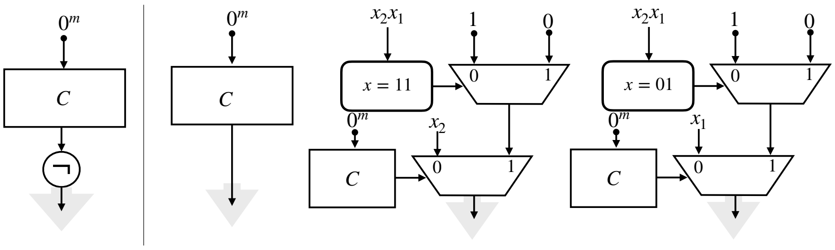


Figure 2: Left: local function of each f_i defined for the proof of $\mathtt{0}^n$ is the fixed point; Right: the first graph (on the left) local function of each f_i for $i \geq 2$, the second (middle) one is the local circuit of f_2, and the last (right) one is the one of f_1, in for the second point of the proof.

each local function if $i \geq 2$ is defined by $f_i(x) = x$, and for the last two, we define f_2 by :

$$f_2(x) = \begin{cases} \mathtt{0} & \text{if } C(\mathtt{0}^n) = 1 \text{ and } x_2x_1 = \mathtt{01} \\ \mathtt{1} & \text{if } C(\mathtt{0}^n) = 1 \\ x_2 & \text{otherwise} \end{cases}$$

We define f_1 similarly, and we represented all the local circuits in Figure 2.

□

As a consequence of Theorem 9, we aim at a statement of the form : any ω-nontrivial FO formula on $\{=, \rightarrow, \trianglelefteq\}$ is P-hard, otherwise it is $\mathcal{O}(1)$.

4.2 FO questions on signature $\{=, \trianglelefteq\}$ are ω-trivial

We will prove that without $\rightarrow$ in the signature, *i.e.* only with an equality relation = and a comparison relation, FO questions are ω-trivial. We will show this for two types of comparison: the bitwise partial order of binary strings denoted $\leq_b$, and the total order of integers denoted $\leq_t$.

Proposition 1. *For any formula ψ on signature $\{=, \trianglelefteq\}$,* **ψ-dynamics** *is ω-trivial.*

By definition of ω-trivial, it is immediate that:

Corollary 1. *For any formula ψ on signature $\{=, \trianglelefteq\}$,* **ψ-dynamics** *can be solved in time $O(1)$.*

We separate the proof in two parts, depending if $\trianglelefteq$ is a partial or a total order.

4.2.1 FO questions on signature $\{=, \leq_b\}$ are ω-trivial

When we are studying FO questions on signature $\{=, \leq_b\}$, the questions do not depend on the dynamic of the automata networks anymore. Hence we prove that:

Proposition 2. *For any formula ψ on signature $\{=, \leq_b\}$, ψ-**dynamics** is ω-trivial.*

We consider structures in the vocabulary $\sigma = \{\subseteq\}$ where $\subseteq$ is a binary relation symbol. The intended interpretation of σ-structures here is finite *Boolean algebras*: that is, $\langle 2^X, \subseteq \rangle$, where X is a finite set, and 2^X is the set of the subsets of X.

We will use the following lemma proved by Libkin:

Lemma 1 ([7, Claim 5.7])**.** *Let $|X|, |Y| \geq 2^k$. Then $\langle 2^X, \subseteq \rangle \equiv_k \langle 2^Y, \subseteq \rangle$.*

We write for each subset $S \subset [\ell]$ with a binary writing such that $S(i) = 1$ if and only if $i \in S$. So we can suppose that lattices on the subsets of $[\ell]$ are actualy on the set $\{0, ..., 2^\ell - 1\}$. Hence $\langle 2^{[\ell]}, \subseteq \rangle \equiv \langle \{0, ..., 2^\ell - 1\}, \leq_b \rangle$ with $\leq_b$ the partial order on binary strings.

Proof of Proposition 2. For the sake of contradiction, let's assume there exists a formula ψ on signature $\{=, \leq_b\}$ that is ω-nontrivial, of quantifier rank k. Then it has an infinite number of models and an infinite number of counter-models, hence models and counter-models of arbitrary sizes. As a consequence there are two graphs G, G' of respective sizes $n, n' \geq 2^k$ such that $G \models \psi$ whereas $G' \not\models \psi$, *i.e.* $G \not\equiv_k G'$, which contradicts Lemma 1. □

4.2.2 FO questions on signature $\{=, \leq_t\}$ are ω-trivial

Proposition 3. *For any formula ψ on signature $\{=, \leq_t\}$, ψ-**dynamics** is ω-trivial.*

We will use the following lemma proved by Libkin:

Lemma 2 ([7, Theorem 3.6])**.** *Let $k > 0$, and let L_1, L_2 be linear orders of length at least 2^k. Then $L_1 \equiv_k L_2$.*

The proof of Proposition 3 from Lemma 2 follows the same principle as the proof of Proposition 2 from Lemma 1.

4.3 Properties

4.3.1 Find a model and a counter-model with exactly one difference

Given a graph G with out-degree 1 for each vertex, and a vertex x we define $G(x)$ the only vertex such that $(x, G(x)) \in E(G)$. Also $\forall i \in \mathbb{N}$ we write e_i the binary number written only with 0 except on the i^{th} bit, with a 1. We show the following proposition for ψ-**dynamics** on FO formula on signature $\{=, \rightarrow, \trianglelefteq\}$.

Proposition 4. *For any ω-nontrivial ψ, there are G, G' equal except on one configuration x such that $G'(x) = G(x) + e_i$ for some $i \in [n]$, and $G \models \psi$ whereas $G' \not\models \psi$.*

Proof. Since ψ is ω-nontrivial, it has infinitely many models and infinitely many counter-models. Moreover, there exists at least one size of graph such that there are two graphs H and H' which verify $H \vDash \psi$, $H' \nvDash \psi$ and $|H| = |H'|$. Otherwise, let's assume toward any contradiction that $\forall n \in \mathbb{N}, \forall H$, graph on n automata $H \vDash \psi$ or $\forall H$, graph on n automata $H \nvDash \psi$. So the validity of ψ only depends on the number of automata, and not the dynamic: indeed given two AN f and f' on n automata, then it is immediate that $\mathcal{G}_f \vDash \psi \iff \mathcal{G}_f \vDash \psi$. Hence there exist a formula φ on signature $\{=, \trianglelefteq\}$ such that ϕ is equivalent to ψ (hence $\forall G \vDash \psi \iff G \vDash \varphi$). According to Proposition 1, φ is ω-trivial, which is a contradiction with the ω-nontriviality of ψ.

We will construct a sequence $H = H_0, H_1, ..., H_k = H'$, for a certain $k \in \mathbb{N}^*$, with $k+1$ distinct graphs ($k \geq 1$ since it is immediate that $H \neq H'$), with the following method, where we assume that $V(H) = V(H')$:

```
$H_0 = H$
$\ell = 0$
While $H' \neq H_\ell$, do:
    Find three configurations $x, y, y'$ such that $x \to y$ in $H_\ell$
    and $x \to y'$ with $y \neq y'$ in $H'$; find $i$ such that $y_i \neq y'_i$, and do:
        $H_{\ell+1} = H_\ell$
        $H_{\ell+1}(y) = H_\ell(y) + e_i$
        $\ell = \ell + 1$
```

The algorithm ends since, we can count the number of differences between H_ℓ and H' with $d_\ell = |\{(y,i) | y \to x \in H_\ell, y \to x' \in H', x_i \neq x'_i\}| \geq 0$, and by construction $d_{\ell+1} = d_\ell - 1$.

Moreover, there exists at least one index j such that $H_j \vDash \psi$ and $H_{j+1} \nvDash \psi$. We take $G = H_j$ and $G' = H_{j+1}$. By construction of the algorithm, G and G' are equal except on one configuration x, such that $G'(x) = G(x) + e_i$ for some $i \in [n]$. □

4.3.2 Permutations and Hamming weight

For every permutation σ on $n \in \mathbb{N}$ elements and for every transition digraph G on n automaton, we define $\sigma(G)$ the digraph such that $\forall v, v \in V(G) \iff \sigma(v) \in V(\sigma(G))$ and $\forall (u,v), (u,v) \in E(G) \iff (\sigma(u), \sigma(v)) \in E(\sigma(G))$.

Lemma 3. *If ψ is a FO-formula on signature $\{=, \to, \leq_b\}$ and σ is a permutation then $\forall G, G \vDash \psi \iff \sigma(G) \vDash \psi$.*

Proof. Let ψ be a FO-formula on signature $\{=, \to, \leq_b\}$, G be a transition digraph on n automaton and $\sigma : \{0,1\}^n \to \{0,1\}^n$ be induced by permutation on n elements.

We show that $G \vDash \psi \implies \sigma(G) \vDash \psi$ (the other implication $\sigma(G) \vDash \psi \implies G \vDash \psi$ is similar, since we just need to consider the permutation σ^{-1} instead).

Since FO-formula on $\{=, \to\}$ are up-to isomorphisms, if ψ does not contain any $\leq_b$, it is immediate. Otherwise, we show that permutations do not change the truth value of a comparison. Indeed, if we write $x = x_1, ..., x_n$ and $y = y_1, ..., y_n$, then we have by definition that $x \leq_b y \iff \forall i, x_i \leq y_i$. Let $\{i_1, ..., i_n\} = [n]$ be such that $\sigma(x) = x_{i_1}, ..., x_{i_n}$, hence we also have $\sigma(y) = y_{i_1}, ..., y_{i_n}$, and since $\forall j, x_{i_j} \leq y_{i_j}$, it means that $\sigma(x) \leq_b \sigma(y)$. □

Contrary to the previous point, swaping two configurations of a graph G with the same Hamming weight to make a graph G', does not imply that $G \vDash \psi \iff G' \vDash \psi$.

$H_k(x)$ is true means that the *Hamming weight* of x is k, *i.e.* $\sum_{i=1}^{n} x_i = k$, when $x = x_1, ..., x_n$. Hence k is the number of `1` in the binary writing of x. For any $k \in \mathbb{N}$, we can define H_k with a FO formula on signature $\{=, \rightarrow, \leq_b\}$, by induction on k.

$$H_0(x) \equiv \forall y, x \leq_b y$$
$$H_{k+1}(x) \equiv \exists y, H_k(y) \wedge y \leq_b x \wedge [\forall z, (y \leq_b z \wedge z \leq_b x) \implies (z = y \vee z = x)]$$

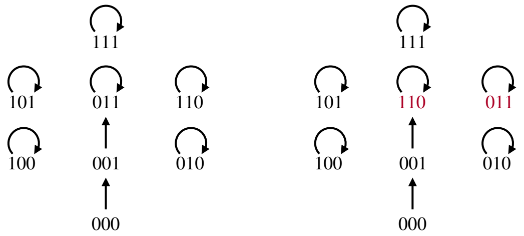


Figure 3: Two transition digraphs with only one difference: a permutation of two configurations of the same Hamming weight (in red). Left: $G \vDash \varphi$. Right: $G' \nvDash \varphi$.

To show that swaping two configurations with the same Hamming weight does not keep the validity of a graph for a formula, we take the following counterexample with $\varphi \equiv \exists x_1, x_2, H_1(x_1) \wedge H_2(x_2) \wedge x_1 \rightarrow x_2$ and Figure 3.

5 FO questions on signature $\{=, \rightarrow, \mathcal{Z}\}$ are P-hard

5.1 General proof

We first resolve a particular case of FO questions on signature $\{=, \rightarrow, \trianglelefteq\}$, where we can only characterize a minimal configurations compared to the cases up to isomorphism (on the signature $\{=, \rightarrow\}$ in particular). We define the unary relation $\mathcal{Z}$ called zero such that for every configuration x, $\mathcal{Z}(x)$ is true if and only if $\exists n \in \mathbb{N}$ such that $x = 0^n$. In particular, with an order $\trianglelefteq$, we have the equivalence: $\mathcal{Z}(x) \iff \forall y, x \trianglelefteq y$.

We write `0` any configuration on the form `0...0` with a length depending on the number of automaton, but we know that it is the only configuration in each graph such that $\mathcal{Z}(x) \iff x = \texttt{0...0}$.

We show the following theorem:

Theorem 10. *For any ω-nontrivial ψ on signature $\{=, \rightarrow, \mathcal{Z}\}$, ψ-**dynamics** is P-hard.*

We recall the following notations as in the article of Gamard, Guillon, Perrot and Theyssier [4]:

Definition 2. *Let G and 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 $\sqcup_z^k G$ denotes $G \sqcup_z ... \sqcup_z G$, with k copies of G. Since the constructions we will sometimes change the configuration of a graph. The configurations are numbered depending on which copy they are in. In the k^{th} copy, we add $k - 1|G|$ to each configuration. We extend this to union of different graphs such that for $G \sqcup_z G'$, the vertices of G are unchanged but in G', we add $|G|$.

Let n be an integer, $\Gamma = (G_1, ..., G_n)$ a n-tuple of graphs, and w a word over alphabet $\{1, ..., n\}$. Define $\mathcal{U}_z^{G,\Gamma}(w)$ by induction as follows: $\mathcal{U}_z^{G,\Gamma}(\epsilon) = G$, and $\mathcal{U}_z^{G,\Gamma}(w_1...w_k) = \mathcal{U}_z^{G,\Gamma}(w_1...w_{k-1}) \sqcup_z G_{w_k}$ (where ϵ is the empty word). We represent $\mathcal{U}_z^{G,\Gamma}(w)$ for $z \in \{1, 2, 3\}$ in Figure 4.

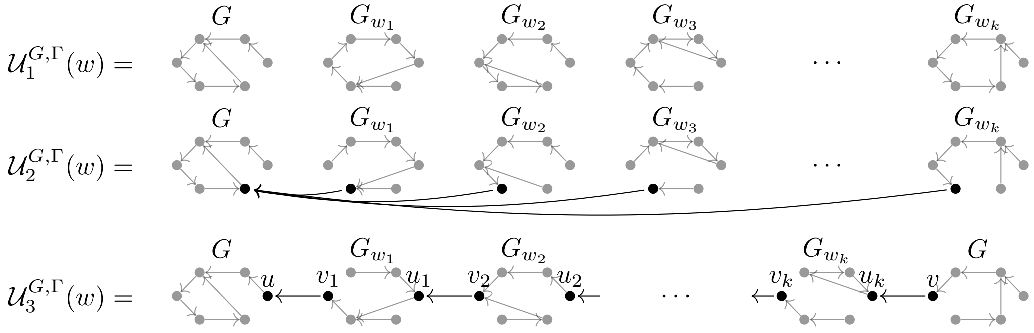

$\mathcal{U}_1^{G,\Gamma}(w) =$
$\mathcal{U}_2^{G,\Gamma}(w) =$
$\mathcal{U}_3^{G,\Gamma}(w) =$
G
G_{w_1}
G_{w_2}
G_{w_3}
G_{w_k}
u
v_1
u_1
v_2
u_2
v_k
u_k
v
...

Figure 4: Illustration of the unions with $\mathcal{U}_z^{G,\Gamma}(w)$ for $z \in \{1, 2, 3\}$ (the pointed nodes are in black while the other parts of the graphs are in grey) [4].

We will first prove that:

Proposition 5. *If ψ is an ω-nontrivial formula on the signature $\{=, \rightarrow, \mathcal{Z}\}$, then there exist nonempty graphs G, G', J, J', and $z \in \{1, 2, 3\}$ such that $|G| = |G'|$, $|J| = |J'|$ and $\forall k \geq 0$, we have $G \sqcup_z (\sqcup_z^k J) \vDash \psi$ and $G' \sqcup_z (\sqcup_z^k J') \nvDash \psi$.*

Preliminary results. 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 $\mathcal{T}$.

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

We prove that with the signature $\{=, \to, \mathcal{Z}\}$ we still have the same lemma, with a similar proof, as the case on $\{=, \to\}$ [7, Lemma 5.2.1]:

Lemma 4. *For all m, the set $\mathcal{E}_m$ is finite.*

Proof. Without loss of generality, we assume that all the formulas are in *prenex form* (quantifiers are at the beginning). So, a formula ϕ is of the form $\mathcal{Q}_1 x_1 ... \mathcal{Q}_m x_m \phi'(x_1, ..., x_m)$, where $\forall i, \mathcal{Q}_i \in \{\exists, \forall\}$ and ϕ' is a quantifier-free formula. A quantifier-free formula $\phi'(x_0, ..., x_{m-1})$ is a Boolean formula over $2m^2 + m$ variables: "$x_i \to x_j$", "$x_i = x_j$" and "$\mathcal{Z}(x_i)$", for $0 \leq i, j < m$. Two Boolean formulas are equivalent if they have the same truth table. There are 2^{2m^2+m} possible assignment for the "variables", thus $2^{2^{2m^2+m}}$ possible truth tables. Consequently, there are at most $2^{m+2^{2m^2+m}}$ nonequivalent formulas of quantifier rank m. Any structure satisfying (resp. falsifying) a formula has to satistfy (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 $w = w_1 w_2 ... w_k$ is a word over $\mathcal{T}$, then $\mathcal{E}_m(w)$ is the word $\mathcal{E}_m(w_1)\mathcal{E}_m(w_2)...\mathcal{E}_m(w_k)$. Similarly, we extend $\mathcal{E}_m$ to multisets over finite words such that if $Y = \{y_, ..., y_n\}$ is a (multi)set of finite words over $\mathcal{T}$, then $\mathcal{E}_m(Y) = \{\mathcal{E}_m(y_1), ..., \mathcal{E}_m(y_n)\}$; and similarly, since graphs can be viewed as multiset of cyclic words, we can define $\mathcal{E}_m(G)$ for any graph G.

Definition 3. *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 symbols $=$ 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. We remark that in a graph $\mathcal{E}_m(G)$ there can be at most one label of $\mathcal{E}_m$ corresponding to an equivalence class of trees containing configuration $\mathtt{0}$, since this configuration is unique in every graph. In each graph we will denote this label $\mathcal{E}_m^{\mathtt{0}}(G)$. We also denote $\mathcal{E}_m^{\mathtt{0}} \subset \mathcal{E}_m$ the set of the equivalence classes whose elements always contain $\mathtt{0}$.

We will need the following theorems, lemma and defintion about partial isomophisms between DULC:

Theorem 11 ([4])**.** *For all m and all graphs G, G', if $\mathcal{E}_m(G) \equiv_m \mathcal{E}_m(G')$ then $G \equiv_m G'$.*

Definition 4. *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).*

Definition 5. *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 $\pi_{H,m} : B_m \to \{0, ..., m \cdot e\} \sqcup \{\omega\}$ 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.*

Lemma 5 (Hanf's lemma [5])**.** *Let m be an integer, and H and H' be DULC. If $\pi_{H,m} \equiv_m \pi_{H',m}$, then $H \equiv_m H'$.*

Theorem 12 ([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 $G \vDash \psi$ if and only if $\mathcal{E}_m(G) \vDash \mathcal{E}(\psi)$.*

We now prove the following lemma, whose proof is inspired by the one of the proposition by Gamard, Guillon, Perrot and Theyssier [4, Proposition 5.2.9.]:

Lemma 6. *In a nonempty DULC H, if there exists a cycle of size at least $e(|\mathcal{E}_m|^e + 1)$ then there exists a nonempty DULC J that doesn't contain any label of $\mathcal{E}^0_m$, such that $\forall k, H \equiv_m H \sqcup (\sqcup^k J)$.*

Proof. If there exists a cycle of size at least $e(|\mathcal{E}_m|^e + 1)$ in H, let call such a cycle $\mathcal{C}$ for counting reasons, there is a word v of length e over the alphabet $\mathcal{E}$ that occurs at least twice in $\mathcal{C}$. Since there is at most one label $\mathcal{E}^0_m(H)$, it cannot appear multiple times, and in particular v does not contain it. From a part of the cycle which contains two occurrences of v and does not contain $\mathcal{E}^0_m(H)$, we construct a cycle J of length at least $e + 1$ (by folding this part on itself at v, such represented on Figure 5). The graph $H \sqcup (\sqcup^k J)$ has the same profile as H for all k; $\pi_{H,m} = \pi_{H \sqcup (\sqcup^k J),m}$.

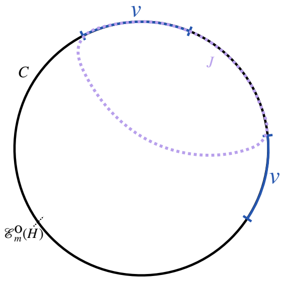


Figure 5: Representation of the construction of J ($\mathcal{E}^0_m(H)$ is not necessarily in the cycle $\mathcal{C}$, but we represent in that case).

According to Lemma 5, it means that for all k, $H \equiv_m H \sqcup (\sqcup^k J)$. □

Definition 6. *A* family *of models is a set of models. For a family of models $\mathcal{F}$ for a formula ψ, we write $\mathcal{E}_m(\mathcal{F}) = \{\mathcal{E}_m(G), \forall G \in \mathcal{F}\}$, hence if we obtain a graph $H \in \mathcal{E}_m(\mathcal{F})$ then there exists a graph $G \in \mathcal{F}$ such that $H = \mathcal{E}_m(G)$.*

We now proceed to a case disjonction, according to some structural property of the family of models and countermodels. For every point of the disjunction, we will use tools from finite model theory. The principle will be to have an unbounded characteristic in models such that we can find a graph that we can copy as many time as we want, and having an easy construction of models as big as we need. After, we will prove how the cases are combined. Theorem 10 will be proved using Proposition 5 and showing that with its conclusion, we can make a L-reduction from **Circuit-Value-Problem**.

Unbounded cycles. We adapt the proof of [4, Proposition 5.3.1.]:

Proposition 6. *If ψ of quantifier rank at most $m \in \mathbb{N}$ has a model G, such that there exists a cycle of size at least $e(|\mathcal{E}_m|^e + 1)$ in $\mathcal{E}_m(G)$, then there exist a nonempty graph J such that $\forall k, G \sqcup (\sqcup^k J) \vDash \psi$.*

Proof. We write $H = \mathcal{E}_m(G)$ and $\phi = \mathcal{E}(\psi)$. Since G is a model of ψ, according to Theorem 12, H is a model of ϕ.

According to Lemma 6, there exists another DULC, J' such that $\forall k, H \equiv_m H \sqcup (\sqcup^k J')$. We note J the graph such that $\mathcal{E}_m(J) = J'$. Also, by definition of $\mathcal{E}_m(\mathcal{F})$, we know that $G \in \mathcal{F}$. Since it is a disjoint union, it is immediate that $\mathcal{E}_m(G \sqcup (\sqcup^k J)) = H \sqcup (\sqcup^k J')$ for all k. By Theorem 11, we have $G \equiv_m G \sqcup (\sqcup^k J)$. Thus, $\forall k, G \sqcup (\sqcup^k J) \vDash \psi$. □

Corollary 2. *If ψ has a an infinite family $\mathcal{F}$ containing models with unbounded cycles, then there exists a graph $G \in \mathcal{F}$ and a non-empty graph J such that $\forall k, G \sqcup (\sqcup^k J) \vDash \psi$.*

Proof. Let ψ be a FO-formula on signature $\{=, \rightarrow, \mathcal{Z}\}$ such that its models have unbounded cycles, and whose quantifier rank is at most $m \in \mathbb{N}$.

Hence, the projection $\phi = \mathcal{E}(\psi)$ also has models with unbounded cycles, and the family $\mathcal{E}_m(\mathcal{F})$ is also an infinite family of models with unbounded cycles. In particular, there exists a model $H \in \mathcal{E}_m(\mathcal{F})$ such that there exists a cycle of size at least $e(|\mathcal{E}_m|^e + 1)$ in it.

Hence the graph $G = \mathcal{E}_m(H)$ verify the hypothesis of Proposition 6, and the result follows. □

Unbounded degrees.

Definition 7. *A* subtree *of a tree T is always* complete, i.e., *spanned by the set of nodes coaccessibles from a given node (the root of the subtree). An* immediate subtree *is a tree whose root has depth 1 in the ambient tree. If T is a tree and $\alpha \in \mathcal{E}_m$, we write $|T|_\alpha$ for the number of immediate subtrees of T of type α.*

We will need the following lemma:

Lemma 7 ([4]). *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 $T \equiv_m T'$.*

We write $deg_G : V(G) \to \mathbb{N}$ the function wich gives the in-degrees of each vertex of a graph G, *i.e.*, if $v \in V(G), deg_G(v) = |\{u \in V(G), (u, v) \in E(G)\}|$. We adapt the proof of Gamard, Guillon, Perrot and Theyssier of [4, Proposition 5.3.3]:

Proposition 7. *If ψ of quantifier rank at most $m \in \mathbb{N}$ has a model G, such that there exists a vertex v of degree $deg_G(v) \geq m \cdot |\mathcal{E}_m| + 1$ then there exist a nonempty graph J such that $\forall k, G \sqcup_2 (\sqcup_2^k J) \vDash \psi$.*

Proof. We know that in the different subtrees with root v in a graph, at most one of them contains $\mathtt{0}$ (hence, whose equivalence class belongs to $\mathcal{E}_m^{\mathtt{0}}$), all the others can be characterized by FO-formulas on $\{=, \rightarrow\}$, and we can copy multiple times these trees without having to duplicate a configuration (here $\mathtt{0}$).

Let $v \in V(G)$ be a vertex such that $deg_G(v) \geq m \cdot |\mathcal{E}_m| + 1$. Hence, v has at least m equivalent immediate subtrees $T_1 \equiv_m ... \equiv_m T_m$ and none of them contains $\mathtt{0}$. We use Lemma 7 on our m equivalent immediate subtrees $T_1, ..., T_m$. It implies that, if we add more copies of T_1 as immediate subtrees of v in G, we have an equivalent graph; *i.e.*, if we put v as the pointed node of G and r the root of T_1 as the pointed node of T_1, then $\forall k, G \equiv_m G \sqcup_2 (\sqcup_2^k T_1)$, and in particular $\forall k, G \sqcup_2 (\sqcup_2^k T_1) \vDash \psi$.

□

Corollary 3. *If ψ has an infinite family $\mathcal{F}$ of unbounded degrees models, then there exists a graph $G \in \mathcal{F}$ and a non-empty graph J such that $\forall k, G \sqcup_2 (\sqcup_2^k J) \vDash \psi$.*

Proof. Let ψ be a FO-formula on signature $\{=, \rightarrow, \mathcal{Z}\}$ such that its models have bounded cycles and unbounded degrees, and whose quantifier rank is at most $m \in \mathbb{N}$.

Since the models of ψ in $\mathcal{F}$ have unbounded degrees, there exist one graph $G \in \mathcal{F}$ and a vertex $v \in V(G)$ such that $deg_G(v) \geq m \cdot |\mathcal{E}_m| + 1$.

With Proposition 7, the result is immediate. □

Unbounded subtree depths or unbounded number of occurrences of a connected component that does not contain $\mathtt{0}$.

Concerning two last cases of the disjunction, for models having unbounded subtree depths or having unbounded number of occurences of a connected component that does not contain $\mathtt{0}$, we provide details in Appendix A (the proofs are similar to the cases above).

Combining the cases.

It is immediate with the following lemma of Gamard, Guillon, Perrot and Theyssier.

Lemma 8 ([4]). *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.*

Combining Lemma 8, with the Corollary 2, Corollary 3, Corollary 4 and Corollary 5:

Proposition 8. *For every FO-formula on signature $\{=, \rightarrow, \mathcal{Z}\}$ with infinite many models, and an infinite family $\mathcal{F}$ of models, there exists a graph $G \in \mathcal{F}$ and a non-empty graph J, and $z \in \{1, 2, 3\}$ such that:*

$$\forall k, G \sqcup_z (\sqcup_z^k J) \vDash \psi.$$

Among the characteristics stated before such as unbounded cycles, unbounded degrees, unbounded hanging tree depths, or unbounded number of occurrences of a connected component that don't contain 0, if we suppose that one of them is true for a family $\mathcal{F}$, we will denote it $\mathfrak{C}$. And for a graph $G \in \mathcal{F}$ we will denote $\mathfrak{C}(G)$ the size of the characteristic in G; hence it can be either the maximum size of a cycle, the maximum degree, the maximum hanging tree depth or the maximum occurences of a connected component in G. We obtain the following properties:

Proposition 9. *For each formula ψ, if it has an infinite number of models in a family $\mathcal{F}$, then there exists a characteristic $\mathfrak{C}$ such that for any constant c we have $\mathcal{F}' \subseteq \mathcal{F}$ with $\forall G \in \mathcal{F}'$, $G \vDash \psi$ and $\mathfrak{C}(G) \geq c$. Furthermore there exist such a constant s such that $\forall G \in \mathcal{F}'$, there exists a nonempty graph J such that $\exists z \in \{1,2,3\}, \forall k, G \sqcup_z (\sqcup_z^k J) \vDash \psi$.*

Proof. First, we know that for all formula there exists an unbounded characteristic $\mathfrak{C}$ for an infinite number of models, according to Lemma 8, in particular for $\mathcal{F}$. We now suppose that $\mathfrak{C}$ is fixed.

We do an induction on $\mathfrak{C}$, given a model $G \in \mathcal{F}$:

- If $\mathfrak{C}$ is the maximum size of a cycle, then according to Proposition 6, $s = e(|\mathcal{E}_m|^e + 1)$ is enough;
- If $\mathfrak{C}$ is the maximum degrees, according to Proposition 7, $s = m \cdot |\mathcal{E}_m| + 1$ is enough;
- If $\mathfrak{C}$ is the maximum hanging tree depth, according to Proposition 11, $s = |\mathcal{E}_m| + 1$ is enough;
- If $\mathfrak{C}$ is the maximum occurences of a connected component in G, according to Proposition 12, $s = m$ is enough.

□

Proof of Proposition 5. ψ is ω-nontrivial. Hence it has an infinite number of models and an infinite number of counter-models. Moreover, $\forall i \in \mathbb{N}, \exists j \geq i$ such that there exist models with j automata and counter-models with j automata. Indeed, the graphs can't depend on the number of automata to be a model or not if ψ is ω-nontrivial.

We first take the family $\mathcal{F}_0^+$ of models of size j such that there exist counter-models of size j, *i.e.* $\mathcal{F}_0^+ = \{G, G \vDash \psi \text{ and } \exists G', G' \nvDash \psi, |G'| = |G|\}$ According to the previous paragraph this family is infinite. Similarly we define $\mathcal{F}_0^-$ for the counter-models.

Moreover, according to proposition 9, for the formula ψ and the family $\mathcal{F}$, there exist a characteristic $\mathfrak{C}$ which is unbounded in $\mathcal{F}$ and a constant s, such that $\forall G \in \mathcal{F}$, $G \vDash \psi$ and $\mathfrak{C}(G) \geq s$ then there exists a nonempty graph J such that $\exists z \in \{1,2,3\}, \forall k, G \sqcup_z (\sqcup_z^k J) \vDash \psi$.

We define $\mathcal{F}^+ \subset \mathcal{F}_0^+$, such that $\forall G \in \mathcal{F}^+, \mathfrak{C}(G) \geq s$. We also define $\mathcal{F}^- \subset \mathcal{F}_0^-$ such that $\forall G \in \mathcal{F}_0^-$, if there exists $G' \in \mathcal{F}^+$ such that $|G| = |G'|$.

According to Proposition 8, there exists $G' \in \mathcal{F}^-$ and a graph J' such that $\forall k, G' \sqcup_z (\sqcup_z^k J') \nvDash \psi$. By construction of $\mathcal{F}^-$, we can also obtain $G \in \mathcal{F}^+$ such that $|G| = |G'|$. By definition of s and $\mathcal{F}^+$, we know that $\mathfrak{C}(G) \geq s$, hence there is a graph J such that $\forall k, G \sqcup_z (\sqcup_z^k J) \vDash \psi$.

In the case where $|J| \neq |J'|$, we can take $\sqcup_z^{|J'|} J$ instead of J and $\sqcup_z^{|J|} J'$ instead of J', and the result is immediate. □

Proof of P-hardness.

Proposition 10. *Let ψ be a formula and z be an element of $\{1,2,3\}$. If there exist nonempty graphs G, G', J, J' such that $|G| = |G'|$, $|J| = |J'|$ and we have $\forall k, G \sqcup_z (\sqcup_z^k J) \vDash \psi$ and $\forall k, G' \sqcup_z (\sqcup_z^k J') \nvDash \psi$, then ψ-dynamics is P-hard.*

Proof. We do a reduction from **Circuit-Value-Problem** with fixed input 0^m. Let C be an instance of **Circuit-Value-Problem**. Depending on the size of the input m of C, we will construct an AN using at least m automata. In particular the number of configuration X will verify: $2^{m-1} < |X| \leq 2^m$.

We know that for the formula ψ, there exist $z \in \{1,2,3\}$ and nonempty graphs G, G', J, J', such that $|G| = |G'|$, $|J| = |J'|$ and we have $\forall k, G \sqcup_z (\sqcup_z^k J) \vDash \psi$ and $\forall k, G' \sqcup_z (\sqcup_z^k J') \nvDash \psi$.

We construct a circuit in Figure 6 that can compute two graphs $G \sqcup_z (\sqcup_z^p J)$ and $G' \sqcup_z (\sqcup_z^p J')$ and that will choose only one of them depending of the output of C. The integer p is choosen such that $2^{m-1} < |G| + p|J| \leq 2^m$ (similarly we have $2^{m-1} < |G'| + p|J'| \leq 2^m$). We suppose that m is big enough so that $|J| \leq 2^{m-1}$, $|G| \leq 2^m$, (hence also $|J'| \leq 2^{m-1}$, and $|G'| \leq 2^m$).

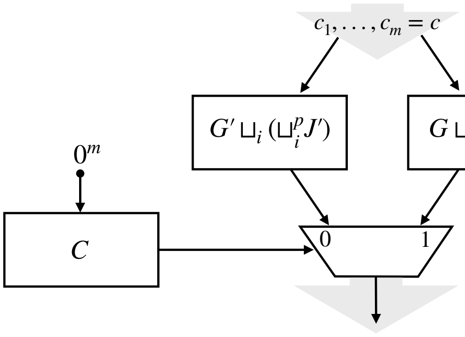


Figure 6: Base circuit (we recall the explanation of the nomenclature is in Section 2.4).

We just need to construct $G \sqcup_z (\sqcup_z^p J)$ and $G' \sqcup_z (\sqcup_z^p J')$. We can use the same construction, and it only depends on z, we give the one for $z = 1$ in Figure 7, while for $z = 2$ or 3 they are given in Annex B.

We call f_G the AN corresponding to the graph G, and f_J for J. The circuit of $G \sqcup_1 (\sqcup_1^p J)$, consist in computing for each input c the function f defined by:

$$\begin{aligned} f(c) &= f_G(c) \text{ if } c < |G| \\ &= f_J(r) + |J| \cdot i + |G| \text{ otherwise, where } c - |G| = |J| \cdot i + r \end{aligned}$$

In the circuit we use the fact that $r = c - |G| \mod |J|$, and $i = \lfloor \frac{c - |G|}{|J|} \rfloor$.

We obtain the fact that for H the graph obtained, $H \vDash \psi \iff C(0^n) = 1$.

Moreover the reduction is in L. Indeed we first notice that the circuits of G and J take a constant space here, since they don't depend on m and we have supposed that

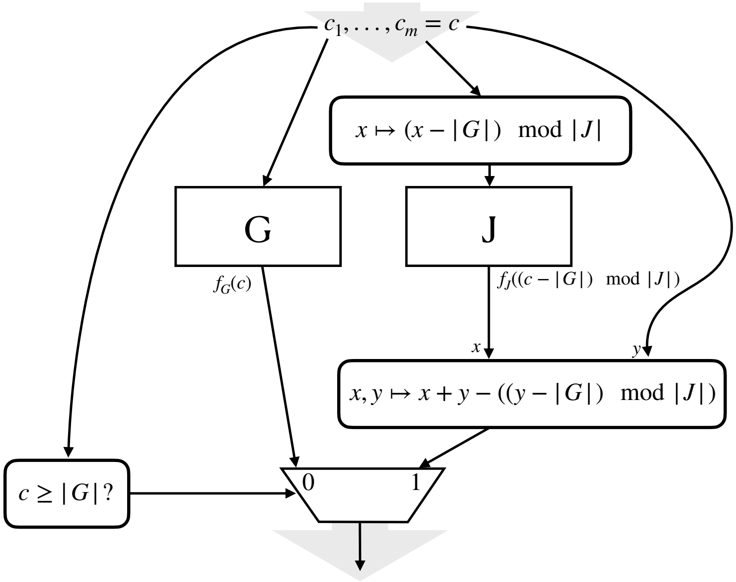

$c_1, \ldots, c_m = c$
$x \mapsto (x - |G|) \mod |J|$
G
J
$f_G(c)$
$f_J((c - |G|) \mod |J|)$
x
y
$x, y \mapsto x + y - ((y - |G|) \mod |J|)$
$c \geq |G|?$
0
1

Figure 7: Circuit of $G \sqcup_1 (\sqcup_1^p J)$.

if their input was taking less than m bits we would troncate it. Also, concerning all the other boxes, they have a constant number of operations depending on the logarithm of the size of the input (depends on the number of bits). Hence the circuit can be constructed in $O(\log(|C|))$ space, depending on the input ℓ. □

Proof of Theorem 10. With Proposition 10 and Proposition 5, it is immediate. □

5.2 Extention

Lemma 9. *Let $\mathcal{Z}_1, \ldots, \mathcal{Z}_k$ be a finite number of unary relations, such that there exist k binary numbers $z_1, \ldots z_k$ such that $\mathcal{Z}_i(x)$ is true if and only if $x = z_i$.*

*For any ω-nontrivial ψ on signature $\{=, \to, \mathcal{Z}_1, \ldots, \mathcal{Z}_k\}$, ψ-**dynamics** is P-hard.*

Proof. It is exactly the same proof as in the case with the signature $\{=, \to, \mathcal{Z}\}$ in Theorem 10. We note here the details that change:

- The model G used in Proposition 5 must be non empty but here we suppose that we have at least $\max\{z_i, i \in [\![k]\!]\}$ configurations, since we consider that all the configurations z_i belong to the graph G in the end;
- We must note that the equivalent classes $\mathcal{E}_m$ are not the same as in the case with zero, but it is still the same principle.
- Lemma 4 is still true, but now there are at most $2^{m+2^{2m^2+mk}}$ nonequivalent formulas of quantifier rank m;
- We note that as in Lemma 6 and Proposition 6, if a word u appears twice in a DULC, then it cannot contain any configuration z_i, it is similar as the fact that it could not contain $\mathtt{0}$ before;

◦ For Proposition 7, Proposition 11 and Proposition 12: nothing changes.

In the end, all the propositions discussed above are sufficient to show all the same propositions as in the case with signature $\{=, \rightarrow, \mathcal{Z}\}$, but with our new signature. □

Remark 2. *For k unary relations $\mathcal{Z}_1, ..., \mathcal{Z}_k$, a formula ψ and a dynamics $\mathcal{G}_f$ on n automata, if $G \vDash \psi$, it depends on the order on the automata. Indeed, for a permutation σ on n elements, we do not necessarily have $\sigma(G) \vDash \psi$. For example with $z_1 =$ `100` and $\varphi = \exists x, x \rightarrow x \wedge \mathcal{Z}_i(x)$. With the left graph in Figure 3, we immediatly see a conterexample if we take $\sigma($`100`$) =$ `001`.*

6 Conclusion

During my internship, I mainly proved that if we add a finite number of unary relations to the signature $\{=, \rightarrow\}$, each one characterising exactly one configuration, then evaluating whether a graph verifies a formula on this signature is either ω-trivial, hence solvable in constant time, or P-hard.

I also found some leads in order to prove our conjecture, that with a signature with an order, partial or total, the complexity is the same: either trivial or P-hard. I showed that we might used Hamming weight since models are not closed under swapping configurations with same Hamming weight; but they are closed under permutations. With the same goal, I showed that we can find two graphs which differ by only one arrow, such that one is a model and the other is a counter-model.

I strongly suspect that a similar method as with the unary relation zero $\mathcal{Z}$, with a disjunction of cases depending on the models, might work, as in the case with a total order; but there are non-trivial issues with the method on models with unbounded cycles which needs to be improved. Lastly, the problem with a bitwise partial order is still open, and it is still not clear whether the same disjunction as the one used previously can still be applied here.

There are still other open questions related to this kind of Rice-like complexity results: does it still hold on a fixed alphabet *i.e.* is it possible to have the same theorems for Boolean AN, and to do a construction fitting graphs whose sizes are povers of two? Indeed, most of our construction have an arbitrary number of configurations, hence it doesn't correspond to any Boolean AN, and the adaptation to this specific case has yet to be found. Is it possible to extend the theorem to monadic second-order formulas ? Or to extend it also to non-deterministic AN ? The main difficulty is that lot of trivial problems for the deterministic case are no longer trivial for the non-deterministic case.

Acknowledgments

I would like to thank my supervisors Kévin Perrot and Enrico Porreca for introducing me to the world of automata networks, and also for helping me improve my understanding about the tools of finite model theory and complexity theory.

I also would like to thank all the other students for their support and all the discussions we had: Isabella, Ricardo, Léah, Florian, Émilie, Marius, Yannis, Marc, Guillaume, and Julien.

References

[1] N. Alon. Asynchronous threshold networks. *Graphs and Combinatorics*, 1(1):305–310, 1985.

[2] F. Bridoux, C. Gaze-Maillot, K. Perrot, and S. Sené. Complexity of limit-cycle problems in boolean networks. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 135–146. Springer, 2021.

[3] P. Floréen and P. Orponen. On the computational complexity of analyzing hopfield nets. *Complex Systems*, 1989.

[4] G. Gamard, P. Guillon, K. Perrot, and G. Theyssier. Rice-Like Theorems for Automata Networks. In *Proceedings of STACS'2021*, volume 187 of *LIPIcs*, pages 32:1–32:17. Schloss Dagstuhl, 2021.

[5] W. Hanf. Model-theoretic methods in the study of elementary logic. In J. W. Addison, editor, *Journal of Symbolic Logic*, pages 132–145. Amsterdam: North-Holland Pub. Co., 1965.

[6] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[7] L. Libkin. *Elements of finite model theory*, volume 41. Springer, 2004.

[8] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[9] P. Orponen. Neural networks and complexity theory. In *International Symposium on Mathematical Foundations of Computer Science*, pages 50–61. Springer, 1992.

[10] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1 edition, 1993.

[11] K. Perrot. *Études de la complexité algorithmique des réseaux d'automates.* PhD thesis, Aix-Marseille Université, 2022.

[12] H. G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the AMS*, 74(2):358—366, 1953.

[13] R. Thomas. Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3):563–585, 1973.

A Last cases of the proof about FO questions on signature $\{=, \rightarrow, \mathcal{Z}\}$

Unbounded subtree depths. In a graph G, a *hanging tree* is a connected component of the graph obtained from G by removing all the edges in cycles. The *treedepth* of a tree T, written $d(T)$ can be define by induction on the number of vertices p of T: if $p = 0$ then its treedepth is 0; otherwise let r be the root of F the set of its immediate subtrees, $d(T) = 1 + \max\{d(T'), \forall T' \in F\}$.

We will prove a lemma similar to Lemma 5.3.4.[4]:

Lemma 10. *If ψ of quantifier rank at most $m \in \mathbb{N}$ has a model G, such that has a hanging tree T of treedepth at least $|\mathcal{E}_m| + 1$, then there exists two subtrees T_1, T_2 of T such that $T_2 \subset T_1$ and $T_1 \equiv_m T_2$.*

Proof. We recall that $\mathcal{E}_m$ is finite according to Lemma 4. We 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. Since, in the tree T has a treedepth bigger than $|\mathcal{E}_m| + 1$, by pigeonhole principle, T admits two nodes with the same label, the first one being an ancestor of the other one. We write T_1 and T_2 the two subtrees whose roots are these two respective nodes. Hence $T_2 \subset T_1$. And by definition of label, $T_1 \equiv_m T_2$. □

We will also need the following lemma:

Lemma 11 ([4])**.** *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 $T \equiv_m T'$.*

Remark 3. *If t' is a subtree of t, the latter being also a subtree of T, and $t \equiv_m t$, it means that t doesn't contain the configuration 0; hence, on the signature $\{=, \rightarrow, \mathcal{Z}\}$, there is no problem to replace t by t'.*

We prove the following proposition, whose proof is similar to the one of Proposition 5.3.6. [4]:

Proposition 11. *If ψ of quantifier rank at most $m \in \mathbb{N}$ has a model G, such that has a hanging tree t of treedepth at least $|\mathcal{E}_m| + 1$, then there exists a nonempty graph J such that $\forall k, \tilde{G} \sqcup_3 (\sqcup_3^k J) \vDash \psi$.*

Proof. t has a treedepth is bigger than $|\mathcal{E}_m| + 1$, according to Lemma 10, there exist two subtree of t: T and T' such that $T' \subset T$ and $T \equiv_m T'$. We write the tree $T'' = T \backslash T'$. Let $G = (\tilde{G} \backslash T') \sqcup T'$ be a disconnected graph. We name two pointed nodes u and v: u is in the $\tilde{G} \backslash T'$ part and is the the leaf of t that should have been the parent of T', and v is the root of T'.

We also give two pointed nodes to each graph T'': u is the leaf that would be the parent of T' in T and v is its root. By Lemma 11, $\forall k, G(\sqcup^k T'') \equiv_m \tilde{G}$. However $\tilde{G} = G \sqcup T''$ so $\forall k, \tilde{G}(\sqcup^k T'') \equiv_m \tilde{G}$. We represent this construction in Figure 8.

Thus with $J = T''$, $\forall k, \tilde{G} \sqcup_3 (\sqcup_3^k J) \vDash \psi$.

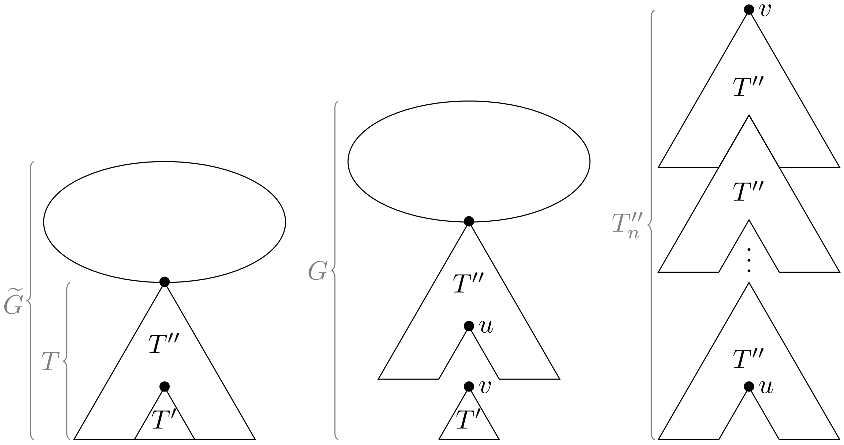


Figure 8: Left: $\tilde{G}$; Middle: G; Right: the union of T'' that we add to G [4].

□

Corollary 4. *If ψ has models with bounded cycles and bounded degrees, and an infinite family $\mathcal{F}$ of unbounded hanging tree depths models, then there exists a graph $G \in \mathcal{F}$ and a non-empty graph J such that $\forall k, G \sqcup_3 (\sqcup_3^k J) \vDash \psi$.*

Proof. Let m be the quantifier rank of ψ. $\mathcal{E}_m$ is finite according to Lemma 4, so there exists a graph $G \in \mathcal{F}$ with a hanging tree depth bigger than $|\mathcal{E}_m| + 1$. According to Proposition 11, the result is immediate. □

Unbounded number of occurrences of a connected component that doesn't contain 0.

We will need the following lemma, in the $\{=, \to\}$ case:

Lemma 12 ([4]). *Let G and J be graphs and m an integer. For all $k, k' \geq m$ we have $G \sqcup (\sqcup^k J) \equiv_m G \sqcup (\sqcup^{k'} J)$.*

Proposition 12. *If ψ of quantifier rank at most $m \in \mathbb{N}$ has a model G, such that more than m connected components that don't contain 0 that are isomorph to a graph J, then $\forall k, G \sqcup (\sqcup^k J) \vDash \psi$.*

Proof. We assume that in a graph G there are more than m connected components that don't contain 0 and that are isomorph to a graph J. Hence there exist $i \geq m$ and a graph H and such that $G = H \sqcup (\sqcup^i J)$.

According to Lemma 12, $\forall k, H \sqcup (\sqcup^i J) \equiv_m H \sqcup (\sqcup^{i+k} J)$, because J . Since $G \vDash \psi$, we have $\forall k, G \sqcup (\sqcup^k J) \vDash \psi$. □

The following corollary is then immediate:

Corollary 5. *If ψ has models with bounded cycles, bounded degrees, and bounded hanging tree depths, and an infinite family $\mathcal{F}$ of unbounded number of occurrences of a connected component that doesn't contain 0 models, then there exists a graph $G \in \mathcal{F}$ and a non-empty graph J where J doesn't contain 0 such that $\forall k, G \sqcup (\sqcup^k J) \vDash \psi$.*

B Circuit for the reduction of signature $\{=, \rightarrow, \mathcal{Z}\}$

We give here the representation of the circuits to construct $G \sqcup_2 (\sqcup_2^p J)$ and $G \sqcup_3 (\sqcup_3^p J)$.

Concerning $G \sqcup_2 (\sqcup_2^p J)$, we construct it by considering that the first $|G|$ configurations simulate G, and every group of $|J|$ configurations after will simulate one copy of J. Hence we remark that computing $c - [(c - |G|) \mod |J|]$ on an input $c \geq |G|$ is computing the smallest configuration of the copy J that contains c. If we note u the pointed node of J and v the one of G, the function of the graph f_2 is defined by:

$$\begin{aligned} f_2(c) &= f_G(c) \text{ if } c < |G| \\ &= f_J((c - |G|) \mod |J|) + c - [(c - |G|) \mod |J|] \text{ otherwise if } c \neq u \\ &= v \text{ if } c = u \end{aligned}$$

We represent the circuit that computes f_2 in Figure 9.

For $G \sqcup_3 (\sqcup_3^p J)$, it is the same principle but we need to take care about u, u', v and v'. u' and v' are the pointed nodes of G and u and v the ones of J. The graph is the dynamics of the function f_3 represented on Figure 10 and defined by:

$$\begin{aligned} f_3(c) &= f_G(c) \text{ if } c < |G| \text{ and } c \neq u \\ &= v + |G| \text{ if } c < |G| \\ &= f_J((c - |G|) \mod |J|) + c - [(c - |G|) \mod |J|] \text{ otherwise if } c \neq u \\ &= v' \text{ if } c + 2|J| + |G| \geq 2^m \text{ and } c = u \\ &= v + c - [(c - |G|) \mod |J|] + |J| \text{ if } c = u \end{aligned}$$

We note that we used the fact that $\lfloor \frac{c-|G|}{|J|} \rfloor \cdot |J| + |G| = c - (c - |G|) \mod |J|$.

C FO questions on signature $\{=, \rightarrow\}$ are P-hard

We know they are hard for NP or coNP, but they are also hard for P. The main improvement here to the proofs of Gamard, Guillon, Perrot and Theyssier [4], is that we show that the reduction they have done can be done with respect to L.

Theorem 13. *For any ω-nontrivial ψ on signature $\{=, \rightarrow\}$, ψ*-**dynamics** *is* P*-hard.*

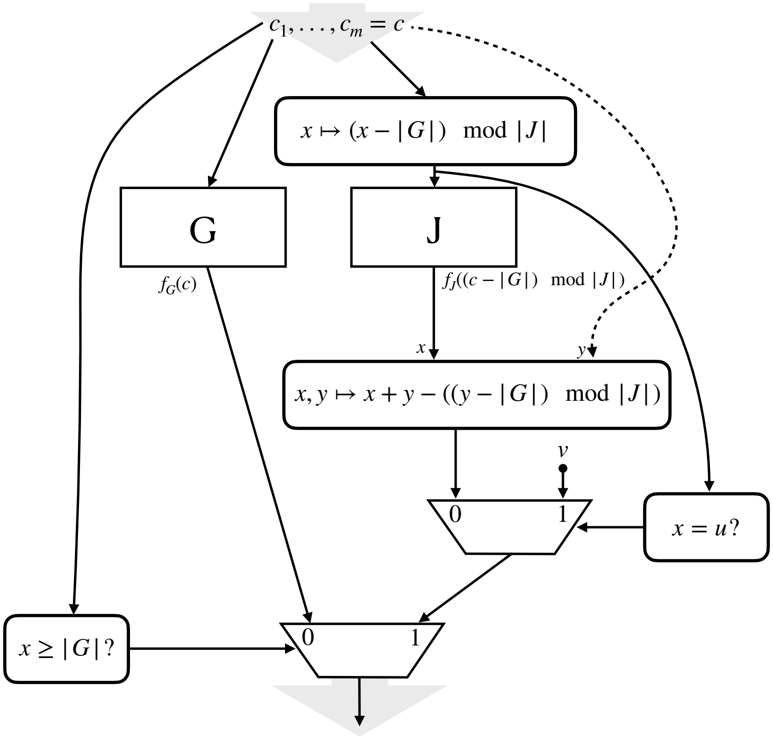

$c_1, \ldots, c_m = c$
$x \mapsto (x - |G|) \mod |J|$
G
J
$f_G(c)$
$f_J((c - |G|) \mod |J|)$
x
y
$x, y \mapsto x + y - ((y - |G|) \mod |J|)$
v
0
1
$x = u?$
$x \geq |G|?$
0
1

Figure 9: Circuit of $G \sqcup_2 (\sqcup_2^p J)$.

We adapt the proof of Gamard, Guillon, Perrot and Theyssier, that with any ω-nontrivial ψ on signature $\{=, \rightarrow\}$, ψ**-dynamics** is NP-hard [4]. Hence we prove proposition 13 and lemma 13.

SAT
Input: a formula φ with n variables
Question: is there an n-tuple $(x_1, x_2, ..., x_n)$ such that $\varphi(x_1, ...x_n)$ is true?

UNSAT
Input: a formula φ with n variables
Question: is $\varphi(x_1, ...x_n)$ is false for every n-tuple $(x_1, x_2, ..., x_n)$?

Proposition 13. *SAT and UNSAT are* P*-hard problems with respect to* $\leq^{\mathsf{L}}$.

Proof. First, we prove it for *SAT*. We know that :

Theorem 14 (Cook's theorem [10])**.** *SAT is* NP*-complete.*

In the proof of Papadimitriou for Cook's theorem, the reductions are made in L since it uses $O(\log n)$ space (with n the size of the input). So the proof also shows that $\forall L \in \mathsf{NP}, L \leq^{\mathsf{L}} SAT$.

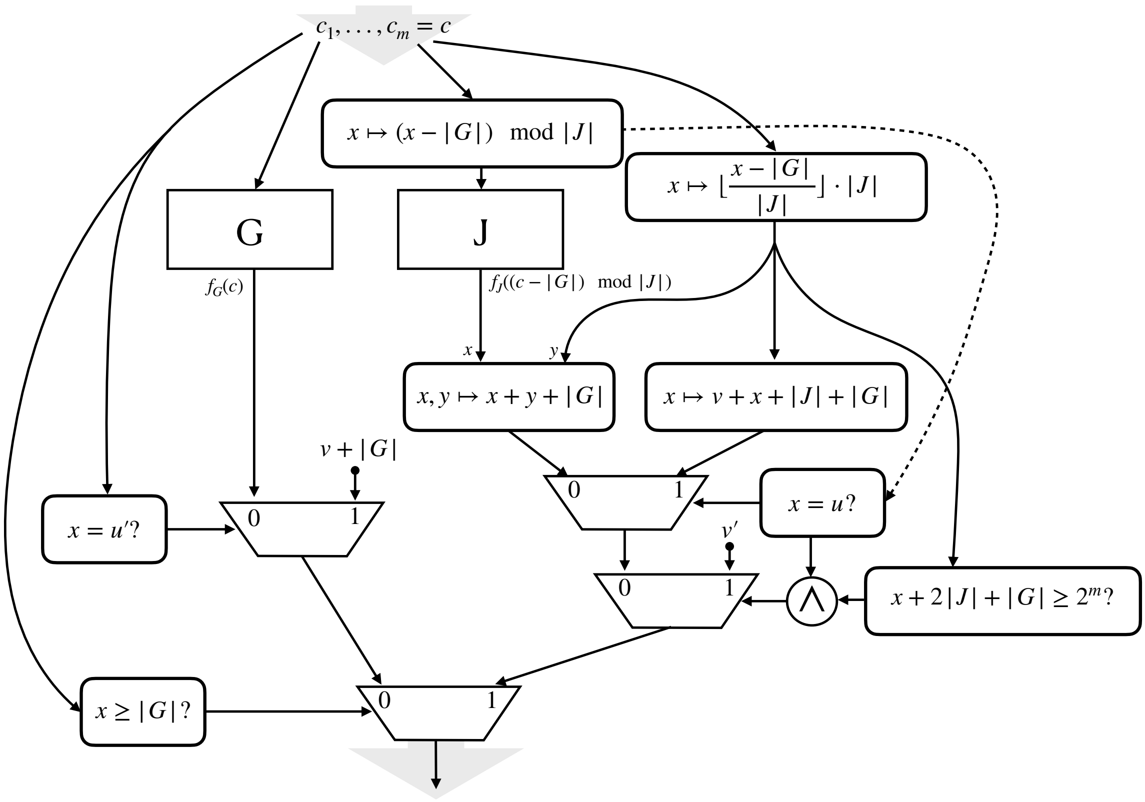

$c_1, \dots, c_m = c$
$x \mapsto (x - |G|) \mod |J|$
$x \mapsto \lfloor \frac{x - |G|}{|J|} \rfloor \cdot |J|$
G
J
$f_G(c)$
$f_J((c - |G|) \mod |J|)$
x
y
$x, y \mapsto x + y + |G|$
$x \mapsto v + x + |J| + |G|$
$v + |G|$
$x = u'?$
0
1
0
1
$x = u?$
v'
0
1
$\wedge$
$x + 2|J| + |G| \geq 2^m?$
$x \geq |G|?$
0
1

Figure 10: Circuit of $G \sqcup_3 (\sqcup_3^p J)$.

In particular, since $\mathsf{P} \subset \mathsf{NP}$, $\forall L \in \mathsf{P}, L \leq^{\mathsf{L}} SAT$. Which means that SAT is P-hard with respect to $\leq^{\mathsf{L}}$.

Concerning $UNSAT$, P is closed under complementation hence it is immediate. □

Lemma 13. *$SAT \leq^{\mathsf{L}}$ ψ-**dynamics** or $UNSAT \leq^{\mathsf{L}} \psi$-**dynamics** .*

To prove lemma 13, we will need the following proposition, which is a direct consequence of Lemma 5.3.9[4] combined with Propositions 5.3.1., 5.3.3., 5.3.6. and 5.3.8[4] of the article *Rice-Like Theorems for Automata Networks* of Gamard, Guillon, Perrot and Theyssier.

Proposition 14 ([4])**.** *Let ψ be a a ω-nontrivial formula on signature $\{=, \rightarrow\}$ and z be an element of $\{1, 2, 3\}$. There exist nonempty graphs G, J, D such that we have:*
- either $\forall k, k' \in \mathbb{N}, G \sqcup_z (\sqcup_z^k J) \vDash \psi$ and $G \sqcup_z (\sqcup_z^k J) \sqcup_z (\sqcup_z^{k'} D) \nvDash \psi$
- either $\forall k, k' \in \mathbb{N}, G \sqcup_z (\sqcup_z^k J) \nvDash \psi$ and $G \sqcup_z (\sqcup_z^k J) \sqcup_z (\sqcup_z^{k'} D) \vDash \psi$.

We will now adapt their reduction to show that:

Proposition 15. *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 $G \sqcup_z (\sqcup_z^k J) \vDash \psi$ and $G \sqcup_z (\sqcup_z^k J) \sqcup_z (\sqcup_z^{k'} D) \nvDash \psi$, then SAT $\leq^{\mathsf{L}}$ ψ-***dynamics**.

We recall the notations of their article, that we already defined in Section 5. If G is a graph k is an integer, and z is in $1,2,3\}$, then $\sqcup_z^k G$ denotes $G \sqcup_z ... \sqcup_z G$, with k copies of G. Now, let n be an integer, $\Gamma = (G_1, ..., G_n)$ a n-tuple of graphs, and w a word over alphabet $\{1, ..., n\}$. Define $\mathcal{U}_z^{G,\Gamma}(w)$ by induction as follows: $\mathcal{U}_z^{G,\Gamma}(\epsilon) = G$, and $\mathcal{U}_z^{G,\Gamma}(w_1...w_k) = \mathcal{U}_z^{G,\Gamma}(w_1...w_{k-1}) \sqcup_z G_{w_k}$ (where ϵ is the empty word).

Let S denote an instance of SAT with s variables. Then $\bar{S}$ 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).

We now prove proposition 15 :

Proof. Let S be an instance of SAT, $z \in \{1,2,3\}$ and G, J, D be graphs such that $|G| < |J| = |D| = \ell$. We will know now construct the graph $G_f = \mathcal{U}_z^{G,(J,D)}(\bar{S}) \sqcup_z (\sqcup_z^k J)$ (with k an integer and an AN f) used in the polynomial reduction [4], but instead copying every graph G, J, D each time we need them, we will only define the AN f, letting the computation for later.

We will define $f : \{0, ..., 2^m\} \to \{0, ..., 2^m\}$ with only one automaton representing the 2^m configuration.

We differenciate multiple cases, whether G is empty or not and depending on the value of z. We will write $f = f_z$ depending on the value of z. We will note $l = \lceil log(\ell) \rceil$.

- First, let's assume that $G = \emptyset$ and $z = 1$ (so we use disjoint union). We will take $m = n + l$. For every $c \in \{0, ..., 2^m\}$, we will note $c_1, ...c_m$ its binary writing. We will define $c_{mod} = c[2^n]$ which represent n automata, whose binary writing is $c_{l+1}, ..., c_m$, and $c_{div} = \lfloor \frac{c}{2^n} \rfloor$ which represent the l other automata, whose binary writing is $c_1, ..., c_l$.

  The goal, is that we will consider for each configuration c that c_{mod} is the number of the graph (J or D) where is c, and c_{div} is a configuration of J or D. To know whether c needs to be in a copy of J or D we need to know the value of $\bar{S}(c_{mod})$. Hence we can define f_1 like this in Figure 11.
- Let's assume that $G \neq \emptyset$ and $z = 1$. We know that the configuration $\ell 0^n$ (in binary, otherwise it is equal to $\ell \cdot 2^n$) is the smallest configuration that is not defined in the case where G is empty (and every higher configuration is not defined too): we want to rename the configuration of G by configurations between $\ell \cdot 2^n$ and $\ell \cdot 2^n + |V(G)| - 1$. Whether $\ell \cdot 2^n + |V(G)| - 1 > l$ or not, we might need to add one automaton (compared to the G empty case) so we have two cases. We will note max $= \ell 0^n$.

  If $\ell \cdot 2^n + |V(G)| - 1 \leq l$, we keep $m = n + l$. We will just associate the configuration

$$
\begin{aligned}
f_1(c) &= f_J(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 1 \\
&= f_D(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 2
\end{aligned}
$$

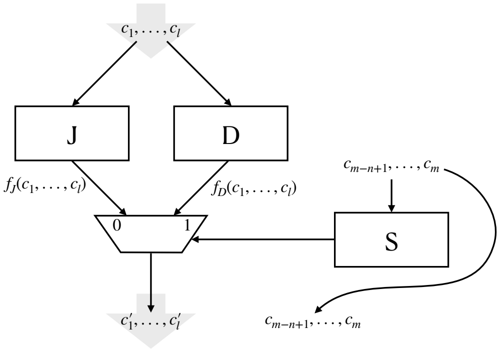


Figure 11: Circuit defined by f_1, where J and D are the circuit of the automata network with the same name, and S the circuit of a SAT instance.

x in G by $\ell \cdot 2^n + x$ (we represent the circuit associated to f_1 in Figure 12)

$$
\begin{aligned}
f_1(c) &= c + f_G(c - \max) \text{ if } \max \leq c < \max + |V(G)| \\
&= f_J(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 1 \\
&= f_D(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 2
\end{aligned}
$$

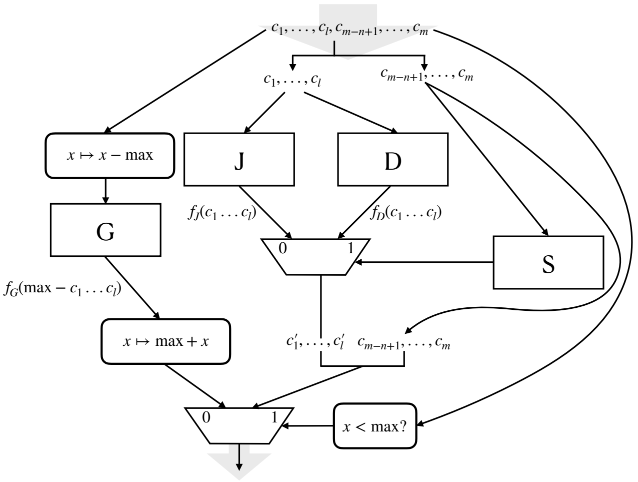


Figure 12: Circuit defined by f_1, where J, D and G are the circuit of the automata network with the same name, and S the circuit of a SAT instance. Here is the representation with $n + l$ automata, and the other case where $\max + |V(G)| \geq 2^{n+l}$ is similar.

Otherwise, if $\ell \cdot 2^n + |V(G)| - 1 > l$, we add one automaton, whose state will be the first bit of the configuration on $m = n + l + 1$ bits. The principle is the same as the previous case.

$$
\begin{aligned}
f_1(c) &= c + f_G(c - \max) \text{ if } \max \leq c < \max + |V(G)| \\
&= \mathtt{0}, f_J(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 1 \text{ and } c_1 = \mathtt{0} \\
&= \mathtt{0}, f_D(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 2 \text{ and } c_1 = \mathtt{0} \\
&= \mathtt{1}, f_G(c_{div}), c_{mod} \text{ if } c_1 = \mathtt{1}
\end{aligned}
$$

Now, for $z = 2$ or $z = 3$, we always assume that G is not empty. Indeed, in the case where G is empty, we can assume that $G = J$ instead (since $J \neq \emptyset$).

- Let's assume that $G \neq \emptyset$ and $z = 2$. As before we note u the pointed node of J and D (since models are up to the isomorphism, we can always rename the vertices to have the same configuration for their pointed node), and u_G the one of G. We define do the same adaptation as in the case $z = 1$, and we keep the same notation. We will do the case where $\ell \cdot 2^n + |V(G)| - 1 > l$ so we use $m = n + l + 1$ automata (the case with $n + l$ automata is the same without the first bit of the configurations). We represent the circuit associate to f_2 in Figure 13.

$$
\begin{aligned}
f_2(c) &= c + f_G(c - \max) \text{ if } \max \leq c < \max + |V(G)| \\
&= \max + u_G \text{ if } c_1 = \mathtt{0} \text{ and } c_{div} = u \\
&= \mathtt{0}, f_J(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 1 \text{ and } c_1 = \mathtt{0} \\
&= \mathtt{0}, f_D(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 2 \text{ and } c_1 = \mathtt{0}
\end{aligned}
$$

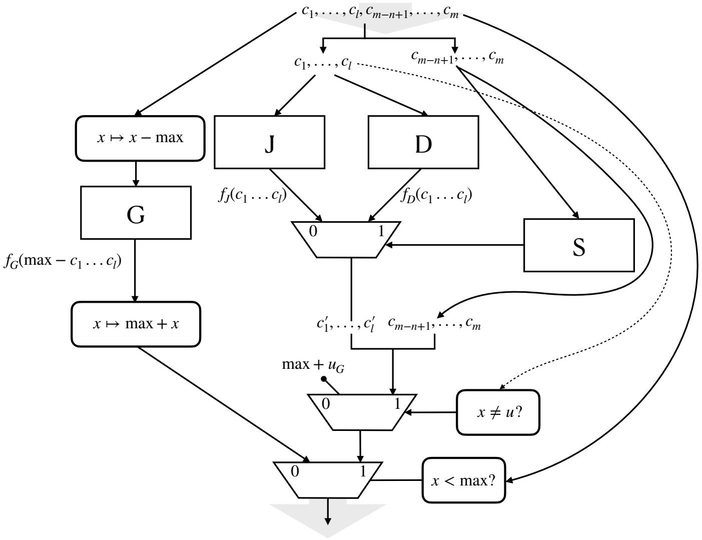


Figure 13: Circuit defined by f_2, where J, D and G are the circuit of the automata network with the same name, and S the circuit of a SAT instance. Here is the representation with $n + l$ automata, and the other case where $\max + |V(G)| \geq 2^{n+l}$ is similar

- Let's assume that $G \neq \emptyset$ and $z = 3$. We note u and v the pointed nodes of G, J and D (since models are up to the isomorphism, we can always rename the vertices to have the same configuration for their pointed node). Here, we assume that $m = n + l + 1$, the case where $m = n + l$ being similar. We represent the circuit associated to f_3 in Figure 14.

$$
\begin{aligned}
f_3(c) &= \mathtt{0}, v, (c_{mod} + 1) \text{ if } c_1 = \mathtt{0} \text{ and } c_{mod} < n \text{ and } c_{div} = u \\
&= \max + v \text{ if } c = \mathtt{0}, u, n \\
&= \mathtt{0}, v, \mathtt{0}^n \text{ if } c = \max + u \\
&= c + f_G(c - \max) \text{ if } \max \leq c < \max + |V(G)| \\
&= \mathtt{0}, f_J(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 1 \text{ and } c_1 = \mathtt{0} \\
&= \mathtt{0}, f_D(c_{div}), c_{mod} \text{ if } \bar{S}(c_{mod}) = 2 \text{ and } c_1 = \mathtt{0}
\end{aligned}
$$

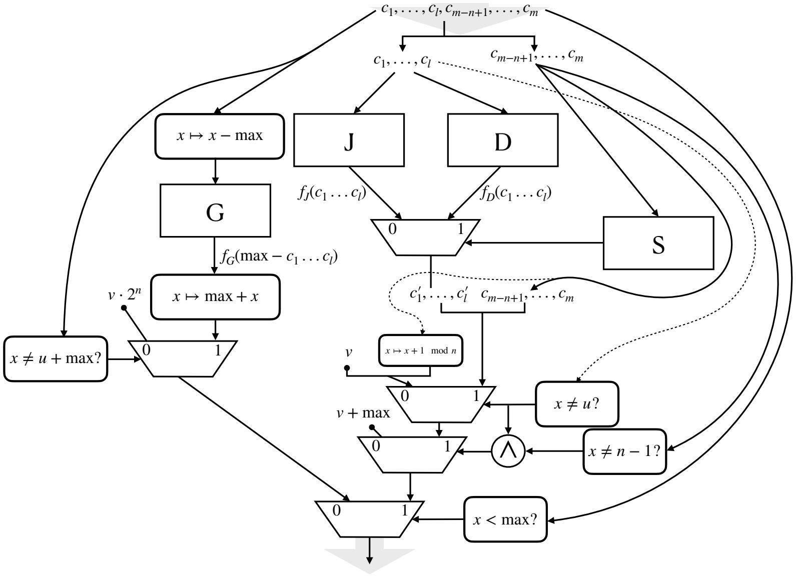


Figure 14: Circuit defined by f_3, where J, D and G are the circuit of the automata network with the same name, and S the circuit of a SAT instance. Here is the representation with $n + l$ automata, and the other case where $\max + |V(G)| \geq 2^{n+l}$ is similar

The reduction is in L. Indeed, the circuits of G, J and D take a constant space here, since they don't depend on m and we have supposed that if their input was taking less than m bits we would troncate it. Also, concerning all the other boxes, they have a constant number of operations depending on the logarithm of the size of the input (depends on the number of bits). Hence the circuit can be constructed in $O(\log(\ell))$ space, depending on the input ℓ. □