

FO Logic on Cellular Automata Orbits Equals MSO Logic

Guillaume Theyssier  

I2M, CNRS, Université Aix-Marseille, France

Abstract

We introduce an extension of classical cellular automata (CA) to arbitrary labeled graphs, and show that FO logic on CA orbits is equivalent to MSO logic. We deduce various results from that equivalence, including a characterization of finitely generated groups on which FO model checking for CA orbits is undecidable, and undecidability of satisfiability of a fixed FO property for CA over finite graphs. We also show concrete examples of FO formulas for CA orbits whose model checking problem is equivalent to the domino problem, or its seeded or recurring variants respectively, on any finitely generated group. For the recurring domino problem, we use an extension of the FO signature by a relation found in the well-known Garden of Eden theorem, but we also show a concrete FO formula without the extension and with one quantifier alternation whose model checking problem does not belong to the arithmetical hierarchy on group \mathbb{Z}^2 .

2012 ACM Subject Classification Theory of computation \rightarrow Models of computation; Theory of computation \rightarrow Logic; Theory of computation \rightarrow Formal languages and automata theory

Keywords and phrases MSO logic, FO logic, cellular automata, domino problem, Cayley graphs

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.154

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version:* <https://hal.science/hal-04558369>

Acknowledgements We thank anonymous referees for their feedback and their suggestions to improve the presentation. We also warmly thank N. Pytheas Fogg for their hints about domino problems on regular graphs and numerous stimulating discussions that inspired this work.

1 Introduction

Symbolic dynamics was historically introduced as the study of one-dimensional infinite words representing discretized orbits of smooth dynamical systems through a finite partition of space [35, 34]. It has since then been largely extended to higher dimensions and arbitrary Cayley graphs of finitely generated groups, and seen rich developments going way beyond the initial motivations. The field of symbolic dynamics would now be better described as the study of sets of configurations (*i.e.* coloring of a graph) which can be defined by local uniform constraints (subshift of finite type, sofic subshift, etc) and maps on configurations acting by a uniform and local update rule (cellular automata, and morphisms between subshifts, which are the continuous maps commuting with translations [26, 12]). A fascinating aspects of these objects is that they are very simple to define, yet can produce very complex behaviors which make them challenging to analyze. They can be considered as a reasonable modeling tool [13], but more importantly, they constitute a natural model of computation for which undecidability and computational hardness can arise in the most simple and seemingly unrelated questions in a spectacular way [28, 30, 25, 9, 29].

In symbolic dynamics, a major trend is to relate the properties of the considered objects (cellular automata or subshifts) to the structure of the underlying graph they are defined on (usually Cayley graphs of finitely generated groups). An emblematic example is the domino problem: since the breakthrough undecidability result of Berger [9], a lot of works



© Guillaume Theyssier;

licensed under Creative Commons License CC-BY 4.0

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).

Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;

Article No. 154; pp. 154:1–154:20



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



have focused on the characterization of Cayley graphs of groups for which the domino problem is undecidable [4, 5, 8, 7] or other graphs with less symmetries [20, 27]. Concerning cellular automata (CA), properties of the global map that can be expressed in first-order (FO) logic (or, said differently, FO properties of their orbit graph) are already challenging. For instance, injectivity and surjectivity problems were shown decidable on \mathbb{Z} [1] and on context-free graphs [36], but undecidable on \mathbb{Z}^2 [29]. Other FO properties of CA were studied in relation to the graph structure: the Gotschalk conjecture [12, 23] asks whether the property “injectivity implies surjectivity” is true for all CA on all group. Besides, the garden of Eden theorem [12] characterizes amenability among finitely generated groups by the fact that another simple FO property¹ is true for all CA on this group.

It turns out that many central problems considered in symbolic dynamics can actually be rephrased in *monadic second order logic* (MSO). It was for instance noticed in [36] for injectivity and surjectivity of CA. MSO is a logical language that has received enormous interest, probably for its balance between expressivity and decidability in many cases. More precisely, for graph properties, algorithmic metatheorems [36, 15, 17] and reciprocals [32, 33] relates the complexity of the MSO model checking problem to the structure of considered graph or graph family. The key parameter at play here is treewidth [39] which is related to the non-existence of arbitrarily large grid minors [40]: bounded treewidth provides positive algorithmic results, while arbitrarily large grid minors often allows lower bounds or undecidability results. For instance, on a Cayley graph of a finitely generated group, the MSO model checking problem is decidable if and only if the group is virtually free (meaning having a free group as finite index subgroup) [33].

Of course, positive algorithmic results for MSO apply to the particular problems studied in symbolic dynamics, but lower bounds or undecidability results for MSO are not directly transferable. For instance, on the graph \mathbb{Z}^2 , undecidability of MSO model checking follows from a straightforward encoding of Turing machines, while undecidability of the domino problem [9] or of injectivity problem of CA [29] requires detailed and non-straightforward constructions that involves ideas and tools of independent interest (aperiodic tile sets and space filling curves for instance). Moreover, it is still open to our knowledge whether both problems are undecidable on any Cayley graph of group which is not virtually free. More generally, much work remains to close the gap between the global understanding of MSO logic on arbitrary graphs and the particular MSO fragments at stake in symbolic dynamics that are mostly understood on Cayley graphs of some group families.

Contributions of the paper. In this paper we explore properties of CA on arbitrary labeled graphs (finite or infinite). To do this, we introduce a definition of local rules of CA that doesn’t use any explicit reference to the local structure of the graph as it is classically done, but instead just relies on the notion of bounded labeled walks and multiset of possible states read at the end of these walks. In particular, we don’t require the graph to be uniform nor to have bounded degree, but our notion exactly corresponds to the classical one on Cayley graphs of finitely generated groups. Moreover, a fixed CA local rule provides a well-defined CA global map on any graph with same label sets. We can thus explore the influence of the graph separately from the local rule. For instance, we can specify a graph property by a CA local rule f and a FO property ϕ of CA orbits: the set of graphs on which the local rule f induces a global map that satisfies ϕ . Our main result (theorems 8, 9 and 12) is then an equivalence between this approach and MSO logic, on arbitrary labeled graphs. Precisely, for every class \mathcal{C} of graphs, the following two conditions are equivalent:

¹ As detailed in Section 5, it uses an additional relation in the FO signature which doesn’t break the main point of our approach, an equivalence with MSO.

- there exists an MSO formula Ψ , such that $G \in \mathcal{C} \iff G \models \Psi$,
- there exists a pair (ϕ, f) , where ϕ is a FO formula and f is a CA local rule, such that $G \in \mathcal{C}$ if and only if the global map induced by f on G satisfies ϕ .

Moreover, the pair (ϕ, f) can be effectively constructed from Ψ , and conversely.

Said differently, (FO,CA) pairs and MSO formulas define exactly the same graph languages, and the corresponding model checking problems are many-one equivalent on any fixed graph. We believe that this new characterization of MSO is particularly relevant in the context of symbolic dynamics.

First, FO properties of orbits of CA are a conjugacy invariant and were much studied as mentioned above. We obtain a characterization of the decidability of FO model checking for CA orbits on Cayley graphs (Corollary 14) and we show that undecidability can be obtained with a fixed FO formula (Corollary 16) exactly on non virtually free f.g. groups, which should be put in perspective with the Ballier-Stein conjecture [6]. We also obtain undecidability of a satisfiability problem for CA on finite graphs for a fixed FO formula (Corollary 17).

Besides, when fixing an arbitrary FO formula and letting the CA vary, we get fragments of MSO that make sense beyond the examples directly motivated by CA theory. In particular, we show in Section 5 that on Cayley graphs of finitely generated groups, such fragments do not depend on the choice of generators, and that the domino problem and its variants (seeded and recurring) are equivalent to the model checking problem of some simple fixed FO formula (Theorem 19). For this we use an additional relation in the case of the recurring domino problem that remains MSO-expressible (Lemma 18). Finally we obtain a FO formula with just one quantifier alternation whose model checking problem does not belong to the arithmetical hierarchy when fixing the graph \mathbb{Z}^2 (Theorem 20).

Warm-up examples. To fix ideas and give some intuition on how FO logic on CA orbits can be used to express graph properties, let us consider two well-known MSO properties and give an informal translation into a pair made of a FO formula and a CA local rule.

► **Example 1 (k -Colorable graphs).** Fix k and consider an undirected graph $G = (V, E)$. Consider the CA local rule with state set $S = \{0, \dots, k-1\}$ such that a vertex in state i changes its state to $i+1 \bmod k$ if it has a neighbor in state i and remains in state i otherwise. It can be checked that the CA induced on G by this local rule has a fixed-point if and only if G admits a proper vertex coloring with k colors (*i.e.* a coloring where no two neighboring vertices have the same color).

► **Example 2 (Connected graphs).** Consider an undirected graph $G = (V, E)$. Consider the state set $S = \{0, 1, a_0, a_1, a_2\}$ and the CA local rule such that a vertex in state $i \in \{0, 1\}$ becomes $1-i$, a vertex in state a_i becomes 0 if it has a neighbor in $\{0, 1\}$ and $a_{i+1 \bmod 3}$ otherwise. We claim that G is connected if and only if the CA induced on G by this local rule has no periodic orbit of minimal period 6, which is obviously an FO property of orbits.

Comparison with another characterization of MSO by automata. Several previous works established equivalence results between logic formalism and automata theory in the context of MSO languages of graphs. As mentioned above, [36] already noticed that some cellular automata properties can be translated into MSO. Following this, [44] and [42] introduced tiling and automata recognizers that are equivalent to (small) fragments of MSO. In [37], alternating distributed graph automata are introduced that recognize exactly the languages of graphs definable in MSO logic. These distributed automata are close to CA in the sense that they run on configurations (coloring of the input graph by states) and use finite local

memory and local communication between neighboring vertices. However, they are highly non-deterministic (alternating) and their accepting mechanism uses both initialization of the run to a particular configuration, and a global knowledge of the final configuration reached (precisely the set of states present in this configuration). Restrictions of this model to deterministic or non-deterministic automata (instead of alternating) gives strictly weaker fragments.

Our goal here is not to define a single automata model equivalent to MSO. Instead our approach motivated by symbolic dynamics uses deterministic CA on one hand and quantifier alternations in a separated FO formula which plays also the role of the accepting condition on the other hand. A key aspect is that we thus get natural fragments of MSO by fixing the FO formula and letting the CA rule vary. This also makes a strong difference with distributed alternating automata in the accepting mechanism since the FO formula does not offer any direct means of initializing some computation on a particular configuration, nor to detect presence of some particular states in a final configuration.

2 Formal definitions

Graphs. A (Σ, Δ) -labeled graph is a graph $G = (V, (E_\delta)_{\delta \in \Delta}, L)$, which can be finite or infinite, where $L : V \rightarrow \Sigma$ is the vertex labeling and $E_\delta \subseteq V \times V$ are the edges labeled by δ . In such a graph, given some finite word $w = w_1 \cdots w_k \in \Delta^*$, a path labeled by w is a sequence of vertices v_1, \dots, v_{k+1} such that, for any $1 \leq i \leq k$, $(v_i, v_{i+1}) \in E_{w_i}$. All graphs considered in this paper are simple, meaning that there is at most one edge of a given label between two given vertices². Such a graph is said to be *connected* if it is connected as an undirected and unlabeled graph, *i.e.* if $G = (V, E)$ is connected where $(v, v') \in E$ if either $(v, v') \in E_\delta$ or $(v', v) \in E_\delta$ for some $\delta \in \Delta$. The set of connected graphs is denoted by \mathcal{C} . Σ will in some case be a singleton and can therefore be silently omitted: we speak about Δ -labeled graph in this case. An important class of graphs studied in symbolic dynamics is that of Cayley graphs of finitely generated (f.g.) groups. Given a f.g. group (Γ, \cdot) and a (finite) set of generators Δ (including their inverses), the associated Cayley graph is the Δ -labeled graph where $(\gamma, \gamma') \in E_\delta$ if and only if $\gamma' = \gamma \cdot \delta$. An undirected graph G_1 is a minor of another undirected graph G_2 if it can be obtained from G_2 by deleting edges and vertices, and by contracting edges (*i.e.* identifying the vertices incident to the edge without creating multiple edges).

Cellular automata. Given a finite set of states S and a set of vertices V , a configuration is an element of S^V . It can be seen as a coloring of vertices by S . A CA is a map from configurations to configurations, that is induced by a uniform local rule. It is generally studied as a dynamical system through its set of orbits, which are sequences of configurations obtained by iterating the map from an initial configuration.

CA are usually defined over a fixed Cayley graph of a (f.g.) group [12]. Following this classical approach, the local rule defining a CA is formally a lookup table and is bound to a particular graph as it relies on local patterns defined over bounded balls of the graph. It also requires the graph to be uniform and of bounded degree. More general definitions were proposed that don't stick to a particular graph [3, 2], but they still rely on the hypothesis of bounded degree and use a particular labeling by port numbers.

² The hypothesis that our graphs are simple will be used in sections 4 and 5. We choose to adopt this hypothesis across the entire paper for simplicity and clarity, however the main results from Section 3 should also hold without this hypothesis.

We introduce in this section a simple definition of CA on arbitrary labeled graphs. The key advantage of our formalism is that a given local rule actually defines a CA on *any* labeled graph for fixed label sets. We can therefore fix a local rule and asks on which graphs the corresponding CA has a given property (as sketched in Section 1): a pair made of a local rule and a property of CA dynamics actually defines a property of graphs. Moreover, on Cayley graphs of f.g. groups our formalism is equivalent to the classical one. The CA local rules we consider can intuitively be seen as finite memory and finite distance exploring machines working as follows in parallel from each vertex v : they walk from v following all possible Δ -labeled walks up to some length r , and harvest states seen at the end of these walks and count their occurrences up to some constant k , then they decide from this information (a multiset) the new state at vertex v . From this point of view, the edge labeling by Δ acts as local directions that give more information on the position in the graph given a labeled walk, while vertex labeling σ gives some level of non-uniformity as in non-uniform cellular automata [18].

Let us now formalize this definition. Given $v \in V$ and a word $w \in \Delta^*$, we denote by $\mathcal{R}^w(v)$ the set of vertices reachable from v by a path labeled by w . If $w = \epsilon$ (empty word), then $\mathcal{R}^w(v) = \{v\}$ by definition. If G is the Cayley graph of a f.g. group, then $\mathcal{R}^w(v)$ is always a singleton, however there are generally several paths reaching the same vertex.

The k -capped multisets over set X are the multisets where no cardinality is greater than k , and are denoted $\text{MS}^k(X) = \{0, \dots, k\}^X$. Given a multiset $m \in \mathbb{N}^X$, we denote by $\text{cap}^k(m)$ the k -capped multiset such that $\text{cap}^k(m)(x) = \max(m(x), k)$. We denote by $A^{\leq r}$ the words of length at most r over alphabet A including the empty word ϵ . Given a set of states S and a configuration $c \in S^V$ and $v \in V$, the k -capped pattern of radius r at v in c is the k -capped multiset $P(c, v, r, k) \in \text{MS}^k(\Delta^{\leq r} \times S)$ defined by:

$$P(c, v, r, k) = \text{cap}^k((w, s) \mapsto \#\{v' \in \mathcal{R}^w(v) : c_{v'} = s\}).$$

► **Definition 3** (CA local rules and global maps). *A CA local rule for (Σ, Δ) -labeled graphs of state set S , radius r and using k -capped multisets ($k \geq 1$) is a map*

$$f : \Sigma \times \text{MS}^k(\Delta^{\leq r} \times S) \rightarrow S.$$

For any (Σ, Δ) -labeled graph with vertices V , the global CA map $F_{G,f} : S^V \rightarrow S^V$ associated to the local map f and graph G is then defined by $F_{G,f}(c)_v = f(\sigma(c_v), P(c, v, r, k))$ for any configuration $c \in S^V$ and any vertex $v \in V$.

In the sequel, when considering a local map f , it always implicitly comes with a specified state set S and values of r and k defining its domain and image sets. We are mainly interested in the case where Σ is a singleton (uniform CA), but incorporating Σ in our definition allows non-uniformity in the local rule as in non-uniform CA [18].

► **Remark 4.** On a Cayley graph of f.g. group, $P(x, v, r, k)$ gives all the information about configuration x restricted to the ball in the graph centered in v and with radius r : indeed, in this case, $P(x, v, r, k)(q, w) = 1$ if and only if the unique vertex $v' \in \mathcal{R}^w(v)$ is such that $x_{v'} = q$, and $\mathcal{R}^w(v)$ describes the entire ball when w enumerates $\Delta^{\leq r}$. From this observation it follows that in the case of Cayley graphs and when Σ is a singleton, the global CA maps from Definition 3 are exactly the classical global CA maps (see for instance [12]).

Definition 3 explains how a given CA local rule f induces a CA global map $F_{G,f}$ on a given graph. $F_{G,f}$ is the main object of study in CA theory, as it represents a dynamical system.

► **Example 5** (Two definitions of Game of Life). Consider the famous Game of Life CA $F : \{0, 1\}^{\mathbb{Z}^2} \rightarrow \{0, 1\}^{\mathbb{Z}^2}$ [10, 22]. In this example Σ is a singleton and thus ignored to simplify notations. First, let G_1 be the Cayley graph of \mathbb{Z}^2 with generators $n = (0, 1)$, $e = (1, 0)$ and their inverses. Let M be the following set of words in $\Delta^{\leq 2}$: $n, n^{-1}, e, e^{-1}, ne, ne^{-1}, n^{-1}e, n^{-1}e^{-1}$. Now define the local rule f_1 of radius 2 and using 4-capped multiset by

$$f_1(\mu) = \begin{cases} 1 & \text{if } \mu(1, \epsilon) = 0 \text{ and } \sum_{w \in M} \mu(1, w) = 3 \\ 1 & \text{if } \mu(1, \epsilon) = 1 \text{ and } 2 \leq \sum_{w \in M} \mu(1, w) \leq 3 \\ 0 & \text{otherwise.} \end{cases}$$

One can check that $F_{G_1, f_1} = F$. Now consider the undirected and unlabeled graph $G_2 = (\mathbb{Z}^2, E)$ with $((i, j), (i', j')) \in E$ if $|i - i'| \leq 1$ and $|j - j'| \leq 1$ and $(i, j) \neq (i', j')$. Here we denote $\Delta = \{u\}$. We then define another local rule f_2 of radius 1 and using 4-capped multiset as follows:

$$f_2(\mu) = \begin{cases} 1 & \text{if } \mu(1, \epsilon) = 0 \text{ and } \mu(1, u) = 3 \\ 1 & \text{if } \mu(1, \epsilon) = 1 \text{ and } 2 \leq \mu(1, u) \leq 3 \\ 0 & \text{otherwise.} \end{cases}$$

One can again check that $F_{G_2, f_2} = F$.

Logics. To make the exposition more concise, we suppose some familiarity with standard concepts of formal logic (variables, assignments, quantification, free variables, etc). MSO logic uses first-order variables (usually denoted by lower-case letters) representing vertices and second-order variables (usually denoted by upper-case letters) representing sets of vertices. To help reading, relations in formulas will use infix notation $(x R y)$ while relation in the meta-language will use the set notation $((x, y) \in R)$.

► **Definition 6** (MSO formulas and their semantics). *The set MSO formulas over label sets (Σ, Δ) is the set of atomic formulas:*

- $x L \sigma$ for x a first-order variable and $\sigma \in \Sigma$ (meaning x has label σ),
- $x E_\delta x'$ for x and x' first-order variables and $\delta \in \Delta$ (meaning that there is an edge labeled δ from x to x'),
- $x = x'$ for first-order variables x and x' (meaning that x is equal to x'),
- $x \in X$ for first-order variable x and second-order variable X (meaning that x belongs to set X),

closed by the usual logic connectives (\vee, \wedge, \neg) and quantifiers (\forall, \exists) . Given an MSO formula Ψ , a (Σ, Δ) -labeled graph G and an assignment α of free variables of Ψ we define the semantics in the standard way starting from the obvious meaning of atomic formula above (see [17] for an in-depth introduction). We write $(G, \alpha) \models \Psi$ when Ψ is true on G with assignment α . If Ψ has no free variable, we simply write $G \models \Psi$ when Ψ is true on G .

We will sometimes use substitution of relations with formulas defining them. For instance we can write $\Psi(X, \Psi_R(x_1, x_2))$, where Ψ is a formula using an additional relation symbol R , to denote the MSO formula obtained by substituting Ψ_R for R in Ψ (with the usual precaution of renaming variables if necessary, see [17]).

We now define FO logic over orbits of CA: they are just formulas allowing quantifications over configurations and using two relations, equality and application of one step of the CA global rule.

► **Definition 7** (FO formulas and their semantics). *The set of FO formulas is made of atomic formulas:*

- $y = y'$ (meaning that configuration y is equal to configuration y'),
 - $y \rightarrow y'$ (meaning that the global CA map leads to y' in one step starting from y),
- and closed under the usual logic connectives and quantifiers. Given a FO formula ϕ , a CA global map $F : S^V \rightarrow S^V$ and an assignment β of free variables of ϕ to configurations from S^V , we write $(F, \beta) \models \phi$ to denote that F satisfies ϕ with assignment β following the obvious semantics of formulas starting from the relations above. When ϕ has no free variable, we just write $F \models \phi$.

For a CA global map $F : S^V \rightarrow S^V$ and a FO formula ϕ with free variables (y^1, \dots, y^n) , we use the shortcut $F \models \phi(c^1, \dots, c^n)$ for configurations $c^1, \dots, c^n \in S^V$ to express that $(F, \beta) \models \phi$ where β is the assignment given by $y^i \mapsto c^i$ for $1 \leq i \leq n$. We will also use the FO shortcut $y^0 \rightarrow_{\neq}^k y^k$ to represent the FO formula expressing that y^0 leads to y^k in k steps and the $k + 1$ configurations involved in this partial orbit are pairwise different:

$$y^0 \rightarrow_{\neq}^k y^k \stackrel{\text{def}}{=} \exists y^1, \dots, \exists y^k : \bigwedge_{0 \leq i < k} y^i \rightarrow y^{i+1} \wedge \bigwedge_{i \neq j} y^i \neq y^j.$$

Notation convention: we will always use letter Ψ for MSO formulas, x or X for MSO variables, α for MSO assignments, ϕ for FO formulas, y for FO variables, G for graphs, f for CA local rules, F for CA global maps and c for configurations. We use notation c_v to denote state of configuration at vertex v , that's why we prefer the exponent notation c^1, c^2, \dots to denote several configurations.

Combining graphs, CA and logics. The above definitions suggest various definitions of sets of objects (or languages): the graph language $\mathcal{G}(\Psi) = \{G : G \models \Psi\}$, the graph language $\mathcal{G}(\phi, f) = \{G : F_{G,f} \models \phi\}$, the set of CA local rules $\mathcal{CA}(\phi, G) = \{f : F_{G,f} \models \phi\}$, where we use the notation convention above, and where Σ and Δ are fixed so graphs are actually (Σ, Δ) -graphs and CA local rules are rules for such graphs. Moreover, $\mathcal{CA}(\phi, G)$ can be seen as decision problems where inputs are given as local maps of CA (model checking problem of ϕ on G).

3 Translation results

3.1 From FO/CA pairs to MSO

Whatever the state set, a CA configuration can be represented as a tuple of vertex sets: we can code the state at a vertex by the number of sets it belongs to among the tuple. This way, FO variables can easily be translated into tuples of second-order MSO variables undergoing the same quantification and we get an onto map from possible assignments of the tuple of second-order MSO variables onto possible assignments of the corresponding FO variable.

Under that coding, equality of configurations translates into a simple MSO formula with just one universal first-order quantifier. It remains to show that the other relation in the signature of FO, relation \rightarrow which represents the application of one step of the CA global map, can also be translated into MSO: this boils down to checking that at each vertex the local rule is correctly applied, which itself boils down to counting up to some constant occurrences of states that can be reached by a labeled walk of bounded length.

► **Theorem 8.** *There is a recursive translation τ from pairs (ϕ, f) made of a FO formula ϕ and a CA local rule f to MSO formulas such that the following equivalence holds for any graph G : $F_{f,G} \models \phi \iff G \models \tau(\phi, f)$.*

3.2 From MSO to FO/CA pairs

This converse translation is less straightforward. Let us first give a simplified overview by considering an MSO formula Ψ in prenex normal form, and describing its translation into a CA local rule f together with a FO formula ϕ .

We will use binary configurations (*i.e.* elements of $\{0, 1\}^V$) to code either second-order variable assignments (a set coded by its indicator function) or first-order variable assignments (a singleton coded by its indicator function). More generally, we can code several variable assignments in a configuration made of several binary components, *i.e.* configurations over a product alphabet $S = \{0, 1\} \times \cdots \times \{0, 1\}$. Given a configuration made of a product of binary components coding an assignment of several variables, the truth of an atomic formula using these variables over this assignment can be checked by a CA local rule in a distributed manner. With slightly more work and using a particular FO property of orbits, we can actually test any quantifier free formula in a distributed manner.

The CA local rule together with the FO formula we are going to construct will essentially enforce an erasing process that starts from a configuration made of a product of binary components (to code an assignment of all MSO variables at once) and then removes components of the product one by one at successive steps until reaching a fixed point: on one hand having all information about variables assignment at the start allows to check the truth of the quantifier-free matrix of the MSO formula as hinted before, and, on the other hand, having components to disappear individually in successive steps allows to make a FO quantification over configurations following exactly the MSO quantification over variables.

For instance, taking MSO formula $\Psi = \forall X_1, \exists x_2, \forall X_3, R(X_1, x_2, X_3)$, we construct a FO formula that is essentially of the form:

$$\forall y_1, \exists y_2, \forall y_3 : y_3 \rightarrow y_2 \rightarrow y_1 \curvearrowright$$

and a CA local rule that will ensure that $y_1 \approx a_1$, $y_2 \approx (a_1, a_2)$ and $y_3 \approx (a_1, a_2, a_3)$ where a_1 is an assignment for X_1 , a_2 is an assignment for x_2 and a_3 is an assignment for X_3 . It will also ensure, when in configuration y_3 , that the assignments (a_1, a_2, a_3) satisfy $R(X_1, x_2, X_3)$. It is important to note that successive choices of assignments of variables y_1 , y_2 and y_3 corresponds, up to a simple product encoding, to successive choices of assignments for variables X_1 , x_2 and X_3 . Non-deterministic choices of successive assignments are possible in this construction because they correspond to going backward in time in the canonical orbit enforced by the FO formula above: there is no contradiction with the determinism of cellular automata.

To turn this overview into a concrete construction, several technical points have to be addressed:

- the simplified behavior described above only works on some well-formed configuration; as usual in CA constructions, we will use local error detection and special error states to mark orbits of bad configurations and distinguish them from good ones: here we use two error states that oscillate with period two in order to ensure that any orbit reaching a fixed point has successfully passed all error detection mechanisms.
- to code first-order variables, binary configurations need to have exactly one vertex in state 1 and the local nature of CA prevents from verifying this (it cannot a priori distinguish a configuration with a single 1 from a configuration with two 1s arbitrarily far away, not to mention the case of non-connected graphs). Our construction handles this through the FO formula to be satisfied using a sibling configurations counting trick combined with a particular behavior of the CA which uses additional layers of states.

The erasing process of the CA mentioned above will therefore take several steps for first-order variables, and only one step for second-order variables.

- checking a quantifier-free formula given an assignment of its variables encoded in a configuration is generally not doable in one step by a CA, especially on non-connected graphs that prevent the CA from communicating between components (think of the example: $x \in X \vee y \in X$); to solve this problem our construction will once again rely on a combination of FO logic over several steps and a particular behavior of the CA.

We first give a solution to these technical problems that works when we restrict to connected graphs. This construction is a little simpler than the general case that we address later, and it has the benefit to induce a better controlled dependence of the FO formula on the MSO formula (an aspect that will turn out to be useful in Section 4).

If Ψ is an MSO formula in prenex normal form with quantifier prefix Q_1, \dots, Q_n , its prefix signature is the word describing the alternations of quantifiers taking into account both the type of quantification and the order of quantified variables. More precisely, it is the word over alphabet $\{\forall, \exists\} \times \{1, 2\}$ obtained as follows: first map each quantifier to the alphabet according to the actual type and order, then remove any repetition of consecutive identical letters.

► **Theorem 9.** *There are two recursive transformations τ_{FO} from MSO formulas to FO formulas and τ_{CA} from MSO formulas to CA local rules such that, for any MSO formula Ψ , the pair made of $\phi = \tau_{FO}(\Psi)$ and $f = \tau_{CA}(\Psi)$ verifies:*

1. for any connected graph G the following equivalence holds: $G \models \Psi \iff F_{G,f} \models \phi$,
2. if Ψ is prenex then ϕ depends only on its prefix signature.

Let $\Psi = \mathbf{Q}_1 x_1^1, \dots, \mathbf{Q}_1 x_1^{k_1}, \mathbf{Q}_2 x_2^1, \dots, \mathbf{Q}_2 x_2^{k_2}, \dots, \mathbf{Q}_n x_n^1, \dots, \mathbf{Q}_n x_n^{k_n}, R(x_1^1, \dots, x_n^{k_n})$ be any MSO formula in prenex normal form where $\mathbf{Q}_1, \dots, \mathbf{Q}_n$ are the n quantifiers types (either \forall or \exists and either first or second-order) forming the prefix signature, variables $x_i^1, \dots, x_i^{k_i}$ are bound by quantifier of type \mathbf{Q}_i and $R(x_1^1, \dots, x_n^{k_n})$ is the matrix of the prenex normal form (i.e. a quantifier free formula). We use this numbering of variables grouped by quantifiers type to obtain a more compact FO formula that only depends on the prefix signature of Ψ . Let's write $R(x_1^1, \dots, x_n^{k_n})$ in disjunctive normal form:

$$R(x_1^1, \dots, x_n^{k_n}) = \bigvee_{1 \leq j \leq d} C_j(x_1^1, \dots, x_n^{k_n})$$

where each clause C_j is a conjunction of terms which are atomic formula or negation thereof using variables $x_1^1, \dots, x_n^{k_n}$. Let \mathcal{O} be the set of i such that \mathbf{Q}_i is a first-order quantifier.

Structure of configurations. For each $1 \leq i \leq n$, let $\omega(i) = |\mathcal{O} \cap \{1, \dots, i\}|$ and define $\lambda(i) = i + 2 \cdot \omega(i)$. As it will become clear below, $\lambda(n)$ denotes the length of an orbit along which n particular configurations will be identified. Configurations along this orbit will use distinct state sets, and $\lambda(i)$ will also denotes the number of layers of the i -th configuration. We first introduce sets S_l for $1 \leq l \leq \lambda(n)$ that will be used to hold variable assignments and translate MSO quantification over variables of first or second-order into FO quantification over configurations. S_l is a product of l layers each of the form $\{0, 1\}^{k_i}$ (variable layer) or $\{1, \dots, k_i\}$ (choice layer) for some i , or $\{0, 1\}$ (control layer). Intuitively, variable layers will hold MSO variables assignments, and choice and control layers are used only for first-order variables as a control mechanism. Sets S_l are precisely defined as follows:

- $S_1 = \{0, 1\}^{k_1}$ and if $1 \in \mathcal{O}$ then $S_2 = S_1 \times \{1, \dots, k_1\}$ and $S_3 = S_2 \times \{0, 1\}$,
- for $1 \leq i < n$, $S_{\lambda(i)+1} = S_{\lambda(i)} \times \{0, 1\}^{k_{i+1}}$ and if $i+1 \in \mathcal{O}$ then $S_{\lambda(i)+2} = S_{\lambda(i)+1} \times \{1, \dots, k_{i+1}\}$ and $S_{\lambda(i)+3} = S_{\lambda(i)+2} \times \{0, 1\}$.

154:10 FO Logic on Cellular Automata Orbits Equals MSO Logic

If $j = \lambda(i) \leq l$ with $i \notin \mathcal{O}$ or $j = \lambda(i) - 2 \leq l$ with $i \in \mathcal{O}$ then the j -th layer of S_l is a *variable* layer, denoted as $V_i(S_l)$, and intuitively represents an assignment for the tuple of variables $x_i^1, \dots, x_i^{k_i}$. For $1 \leq j \leq k_i$, we denote by $V_i^j(S_l)$ the j -th binary component of $V_i(S_l)$ which intuitively represents an assignment for variable x_i^j . For $i \in \mathcal{O}$ and $j = \lambda(i) - 1 \leq l$, the j th layer of S_l is a choice layer, denoted $\chi_i(S_l)$. Other layers, precisely j -th layers with $j = \lambda(i)$ with $i \in \mathcal{O}$, are *control* layers and denoted $K_i(S_l)$. Choice layer χ_i together with control layer K_i are used to ensure that the corresponding variable layer V_i correctly encodes a k_i -tuple of assignments of first-order variables, *i.e.* is a k_i -tuple of binary configurations each having exactly one position in state 1.

We denote by π the natural projection from S_l onto S_{l-1} (for any $2 \leq l \leq \lambda(n)$) which removes the last (l -th) layer of elements of S_l .

A well-formed configuration where all vertices are in a state from $S_{\lambda(n)}$ intuitively contains an assignment for all variables involved in formula Ψ (provided the control mechanism for first-order variables to be detailed below has been successful). We need to implement a distributed check of the truth of quantifier-free formula R on such an assignment. The key is to ensure that some clause C_j from the disjunctive normal form of R is chosen uniformly on the entire graph and to check everywhere that each terms of C_j is locally correct given the assignment. For $1 \leq j \leq d$, let $T_j = S_{\lambda(n)} \times \{j\}$ (recall that d is the number of clauses in the disjunctive normal form of $R(x_1^1, \dots, x_n^{k_n})$). We again use notation π to denote the natural projection from T_j onto $S_{\lambda(n)}$, which removes the last component of states. T_j sates will be used to check clause C_j .

We can now define the state set of the CA $\tau_{CA}(\Psi)$ as

$$S = \{e_0, e_1\} \cup \bigcup_{1 \leq l \leq \lambda(n)} S_l \cup \bigcup_{1 \leq j \leq d} T_j$$

where e_1 and e_2 are distinct elements from the rest of S . We say that the *type* of an element of S is l if it belongs to S_l , *error* if it is e_0 or e_1 and *j -truth-check* if it belongs to T_j . We also naturally extend the notation V_i , χ_i and K_i for any state $s \in T_j$ by $V_i(s) = V_i(\pi(s))$, $\chi_i(s) = \chi_i(\pi(s))$ and $K_i(s) = K_i(\pi(s))$.

A configuration $c \in S^V$ is *valid* if the following conditions hold:

- states of all pairs of neighboring vertices of G are of the same type, and not of error type;
- choice layer χ_i of all pairs of neighboring vertices of G are equal;
- at each vertex v the control layers have zeros where the corresponding variable layers indicated by the choice layers have, precisely: $K_i(c_v) \leq V_i^{\chi_i(c_v)}(c_v)$ for all i such that $K_i(c_v)$ is defined (intuitively, a 1 in a control layer is authorized only if there is a 1 in the 'chosen' component of the corresponding variable layer);

Note that this definition of validity is purely local. For μ a capped multiset (second argument of the local rule of a CA), we write $\text{valid}(\mu)$ to express the local validity conditions above on μ : the first two items are checked on each pair of states (s, s') such that $\mu(s, \epsilon) \geq 1$ and $\mu(s', \delta) \geq 1$ for some $\delta \in \Delta$; the third item is checked on state s such that $\mu(s, \epsilon) \geq 1$.

CA local rule. The behavior of the CA local rule $f = \tau_{CA}(\Psi)$ can intuitively be described as follows:

- check the local validity of the configuration and if not generate states of error type that alternate with period 2 (between e_0 and e_1),
- apply projection π on states of type l with $l \geq 2$ and let states of type 1 unchanged,
- on states of type j -truth-check, verify that the assignment of variables $x_1^1, \dots, x_n^{k_n}$ coded in variable layers V_i^j are such that $C_j(x_1^1, \dots, x_n^{k_n})$ holds, and apply π if it is the case, or generate an error state otherwise.

Of course, the behavior on states of type j -truth-check above has to be understood locally since we are defining a CA. The implementation of this distributed truth check is as follows: each term $t(x_a^b, x_p^q)$ appearing in clause C_j (an atomic formula or its negation), where x_a^b is a first-order variable, is checked only at any vertex having a V_a^b component at 1, otherwise it is considered true by default. More precisely, for a pair (σ, μ) made of a vertex label and a capped multiset (arguments of the local rule f), we write $(\sigma, \mu) \models_{loc} C_j$ (clause C_j is locally valid) if the local state (unique s such that $\mu(s, \epsilon)$) is of type j -truth-check and the neighboring states also, and if all terms of C_j are locally true according to the previous rule. It turns out that $(\sigma, \mu) \models_{loc} C_j$ can be checked by a local rule of CA of radius 1 and using 1-capped multisets (*i.e.* sets). Precisely, all possible terms of clause C_j are treated as follows (denoting again s the unique state such that $\mu(s, \epsilon) \geq 1$):

- $x_a^b L \sigma'$ (resp. its negation) is true if and only if $V_a^b(s) = 0$ or if $\sigma = \sigma'$ (resp. $\sigma \neq \sigma'$),
- $x_a^b E_\delta x_p^q$ (resp. its negation) is true if and only if $V_a^b(s) = 0$ or if $1 \in \{V_p^q(q') : \mu(q', \delta) \geq 1\}$ (resp. 1 does not belong to this set),
- $x_a^b = x_p^q$ (resp. its negation) is true if and only if $V_a^b(s) = 0$ or $V_p^q(s) = 1$ (resp. $V_p^q(s) = 0$),
- $x_a^b \in x_p^q$ (resp. its negation) is true if and only if $V_a^b(s) = 0$ or $V_p^q(s) = 1$ (resp. $V_p^q(s) = 0$).

For t a term, we write $(\sigma, \mu) \models_{loc} t$ if t is locally true according to the above definition. Recall that x_a^b is a first order variable and the rest of the construction will ensure that, on configurations that matter, there will always exist a node at which $V_a^b(s) = 1$ so these tests will actually check that the assignments of variables encoded in the configuration do satisfy the term as desired.

The CA local rule f is then defined as follows (denoting again s the unique state such that $\mu(s, \epsilon) \geq 1$):

$$f(\sigma, \mu) = \begin{cases} e_{1-i} & \text{if } s = e_i \\ e_0 & \text{otherwise, and if } \neg\text{valid}(\mu), \\ e_0 & \text{otherwise, and if } s \text{ has type } j\text{-truth-check and } (\sigma, \mu) \not\models_{loc} C_j, \\ s & \text{otherwise, and if } s \text{ is of type 1,} \\ \pi(s) & \text{otherwise.} \end{cases}$$

FO formula. Most of the task of the FO formula is to check that n configurations are well-positioned in an orbit leading to a fixed-point. However, along this orbit, we also have to make checks to ensure that layers corresponding to first-order variables are well formed. For $i \in \mathcal{O}$, the CA behavior already ensures (by generating error states if not) that choice layers χ_i are uniform and that control layers K_i are upper-bounded by the chosen corresponding variable layer V_i^j . In this context, the check is done as follows (intuitively, variable y^i represents a configuration of type $\lambda(i)$ at each vertex, for some $i \in \mathcal{O}$):

$$\text{goodFOVAR}(y^i) \stackrel{\text{def}}{=} \forall y, \forall y', (y^i \rightarrow_{\neq}^2 y' \wedge y \rightarrow_{\neq}^2 y') \Rightarrow \#\text{siblings}(y) = 1,$$

where the formula $\#\text{siblings}(y) = 1$ expresses that there is exactly 1 configuration other than y with same image as y and can be written explicitly in FO as follows:

$$\exists y_s, \exists y_+, y_s \rightarrow y_+ \wedge y \rightarrow y_+ \wedge y_s \neq y \wedge (\forall y' : y' \rightarrow y_+ \Rightarrow (y' = y \vee y' = y_s)).$$

The idea is that a variable layer V_i is good if, for any choice j made in choice layer χ_i , there are only 2 possible ways to correctly complete the control layer K_i , because there is exactly one vertex v at which V_i^j is 1 and therefore at which K_i can be freely chosen to be 0 or 1.

154:12 FO Logic on Cellular Automata Orbits Equals MSO Logic

Let us now define formulas to deal with the global structure of the orbit leading to a fixed point:

$$\text{seq}_1(y) \stackrel{\text{def}}{=} \begin{cases} y \rightarrow y & \text{if } 1 \notin \mathcal{O}, \\ \exists y^0, y \rightarrow_{\neq}^2 y^0 \wedge y^0 \rightarrow y^0 \wedge \text{goodFOVAR}(y) & \text{if } 1 \in \mathcal{O}, \end{cases}$$

and for any $2 \leq i \leq n$

$$\text{seq}_i(y, y^+) \stackrel{\text{def}}{=} \begin{cases} y^+ \rightarrow y & \text{if } i \notin \mathcal{O} \\ y^+ \rightarrow_{\neq}^3 y \wedge \text{goodFOVAR}(y^+) & \text{if } i \in \mathcal{O} \end{cases}.$$

Then, denote by $\text{good}_i(y^1, \dots, y^i)$ for each $1 \leq i \leq n$ the formula:

$$\text{good}_i(y^1, \dots, y^i) \stackrel{\text{def}}{=} \text{seq}_1(y^1) \wedge \bigwedge_{2 \leq k \leq i} \text{seq}_k(y^{k-1}, y^k).$$

The truth check for R has to make a non-deterministic choice of clause C_j to then use the distributed truth check implemented in the CA, we therefore define the following formula to be used in ϕ : $\text{truth}(y) \stackrel{\text{def}}{=} \exists y' \rightarrow y$ which make sense when y represents a configuration everywhere of type $\lambda(n)$.

As we show later, formula $\text{good}_i(y^1, \dots, y^i)$ paired with CA F ensures that the configuration assigned to y^i is a well-formed configuration of type $\lambda(i)$ that holds an assignment for variables $(x_i^1, \dots, x_i^{k_i})$ through its variable components $V_i^1, \dots, V_i^{k_i}$. We use these formulas in ϕ to make restricted domain FO quantifications that exactly correspond to well-formed configurations that hold assignments of the corresponding MSO variables.

To make the formula ϕ more readable, we use the following syntactic sugar to express restricted domain quantification. If ϕ_D and ϕ are formulas containing y as free variable, then:

- $\exists y \in \phi_D, \phi$ stands for $\exists y, \phi_D \wedge \phi$,
- $\forall y \in \phi_D, \phi$ stands for $\forall y, \phi_D \Rightarrow \phi$.

We can finally define FO formula $\phi = \tau_{FO}(\Psi)$:

$$\phi \stackrel{\text{def}}{=} \mathbf{Q}'_1 y^1 \in \text{good}_1(y^1), \mathbf{Q}'_2 y^2 \in \text{good}_2(y^1, y^2), \dots, \mathbf{Q}'_n y^n \in \text{good}_n(y^1, \dots, y^n), \text{truth}(y^n)$$

where the FO quantifier \mathbf{Q}'_i is \exists if the MSO quantifier \mathbf{Q}_i is existential, and \forall if \mathbf{Q}_i is universal.

Correctness of the construction. First, it can be checked that ϕ only depends on the prefix of Ψ and not on R . Second, the construction of f and ϕ are clearly computable from Ψ . The proof of Theorem 9 then relies on two lemmas. The first one ensures that $\text{good}_i(\dots)$ predicates correctly translate assignments of FO variables quantified in ϕ into assignments of MSO variables quantified in Ψ and conversely.

► **Lemma 10.** *Let G be a (Σ, Δ) -labeled graph which is connected. Consider configurations c^1, \dots, c^i with $1 \leq i \leq n$ such that $F_{G,f} \models \text{good}_i(c^1, \dots, c^i)$, then the following holds:*

1. c^k is of type $\lambda(k)$ at each vertex, for $1 \leq k \leq i$;
2. for $1 \leq l \leq i$ and $1 \leq j \leq k_l$, variable component $V_l^j(c^k)$ is the same for all k with $l \leq k \leq i$, and it is such that exactly one vertex is in state 1 when $l \in \mathcal{O}$;

3. if $i < n$, for any assignment α of variables $(x_{i+1}^1, \dots, x_{i+1}^{k_{i+1}})$ there exists c^{i+1} such that

$$F_{G,f} \models \text{good}_{i+1}(c^1, \dots, c^{i+1})$$

and $V_{i+1}^j(c^{i+1}) = \alpha(x_{i+1}^j)$ for all $1 \leq j \leq k_{i+1}$.

From the above lemma, if $F_{G,f} \models \text{good}_i(c^1, \dots, c^i)$ then c^i codes an assignment for all MSO variables x_l^j for $1 \leq l \leq i$ and $1 \leq j \leq k_l$ through components V_l^j of c^i . Precisely, when $l \in \mathcal{O}$ variable x_l^j is assigned to the unique vertex v such that $V_l^j(c_v^i) = 1$, and when $l \notin \mathcal{O}$ variable x_l^j is assigned to the set of vertices $\{v : V_l^j(c_v^i) = 1\}$. We denote this assignment by α_{c^i} . Moreover, α_{c^i} is an extension of assignment α_{c^j} for any $1 \leq j < i$.

Next lemma ensures that $\text{truth}(\dots)$ predicate correctly codes truth of formula $R(\dots)$ (matrix of Ψ) through the previous assignment translation.

► **Lemma 11.** *Under the hypothesis of Lemma 10, it holds that $F_{G,f} \models \text{truth}(c^n)$ if and only if $(G, \alpha_{c^n}) \models R(x_1^1, \dots, x_n^{k_n})$.*

The proof of Theorem 9 then consists in applying inductively the definition of truth by assignments of variables simultaneously in Ψ and ϕ , use Lemma 11 as base case and Lemma 10 for the induction step to translate assignments between MSO variables and FO variables.

Generalizing to arbitrary graphs. In the previous construction, we use the fact that considered graphs are connected in two places: to ensure uniformity of choice layers χ_i and to ensure a uniform choice of j for testing clause C_j with states of type j -truth-check. When generalized to possibly disconnected graphs, such uniformity conditions cannot be checked by the CA alone, simply because the CA has no possibility to communicate between connected components. We can compensate this impossibility by slightly changing the behavior of F and adding new FO constraints in the definition of ϕ . The price to pay is that the new definition of ϕ will depend on all parts of Ψ , not only its prefix signature.

First, the case of choice layers χ_i can be solved easily by de-grouping variables x_i^1 to $x_i^{k_i}$, *i.e.* renumbering variables in the prefix of Ψ by letting $k_i = 1$ and taking $\sum_{1 \leq i \leq n} k_i$ as the new value of n . Then, choice layers become trivial (they contain just one state) and are therefore always uniform by definition.

To solve the case of truth check, we introduce a general pre-image counting trick to ensure that a configuration c is uniform by a FO property. First, each state s of the alphabet S used by c is associated to a distinct prime number p_s , and there is a probing mechanism that selects a set of vertices and allows exactly p_s predecessors at selected vertices which are in state s , and only 1 at vertices which are not selected. The trick to check that c is s -uniform then consists in counting the number of pre-images up to $\max_{s' \in S} p_{s'}$: whatever the set of selected vertices, it should always be a power of p_s .

► **Theorem 12.** *There are two recursive transformations τ_{FO} from MSO formulas to FO formulas and τ_{CA} from MSO formulas to CA local rules such that for any MSO formula Ψ , and any graph G the following equivalence holds: $G \models \Psi$ if and only if $F_{G, \tau_{CA}(\Psi)} \models \tau_{FO}(\Psi)$.*

Using notations from the construction of Theorem 9, let us now describe precisely the modifications required to generalize to arbitrary graphs. As explained above, we assume $k_i = 1$ for all $1 \leq i \leq n$, so choice layers χ_i are always trivial and uniform. States of type l for $1 \leq l \leq \lambda(n)$ are identical as in the construction of Theorem 9. However, we need additional states to implement the truth check for the matrix R of formula Ψ , and it will spread over 3 time steps of the CA. The key is to ensure that some clause C_j from the disjunctive normal form of R is chosen uniformly on the entire graph and to check everywhere

154:14 FO Logic on Cellular Automata Orbits Equals MSO Logic

that each terms of C_j is locally correct given the assignment. Let $2 = p_1 < p_2 < \dots < p_d$ be the first d prime numbers. For $1 \leq j \leq d$, let $T_j^0 = S_{\lambda(n)} \times \{j\}$ and $T_j^1 = T_j^0 \times \{0, 1\}$ and $T_j^2 = T_j^1 \times \{1, \dots, p_j\}$. T_j^1 is used to *mark* vertices, while T_j^2 is used to alter the number of pre-images depending on j and the mark. We use the notation π to denote at the same time the natural projection from T_j^0 onto $S_{\lambda(n)}$, or from T_j^1 onto T_j^0 or from T_j^2 onto T_j^1 , which removes the rightmost component of states. T_j^0 states will be used to check clause C_j , while states from T_j^1 and T_j^2 will be used to guarantee through a pre-image counting trick that the same choice of j is made on the entire graph, thus ensuring correctness of the truth check of formula R .

We can now define the state set of the CA local rule $\tau_{CA}(\Psi)$ as

$$S = \{e_0, e_1\} \cup \bigcup_{1 \leq l \leq \lambda(n)} S_l \cup \bigcup_{1 \leq j \leq d} T_j^0 \cup T_j^1 \cup T_j^2.$$

We say that the *type* of an element of S is l if it belongs to S_l , *error* if it is e_0 or e_1 and (j, m) -*truth-check* if it belongs to T_j^m for $1 \leq j \leq d$ and $m = 0, 1$ or 2 .

A configuration $c \in S^V$ is *valid* if the following conditions hold:

- states of all pairs of neighboring vertices of G are of the same type, and not of error type;
- at each vertex v the control layers have zeros where the corresponding variable layers indicated by the choice layers have, precisely: $K_i(c_v) \leq V_i^{X_i(c_v)}(c_v)$ for all i such that $K_i(c_v)$ is defined (intuitively, a 1 in a control layer is authorized only if there is a 1 in the 'chosen' component of the corresponding variable layer);
- for a state $(s, m, w) \in T_j^2$ where $s \in T_j^0$, $m \in \{0, 1\}$ and $w \in \{1, \dots, p_j\}$, it must be the case that $w = 1$ whenever $m = 0$ (this condition expresses intuitively, that only marked vertices can generate pre-images and it will allow through a pre-image counting trick in the FO formula to ensure that the choice to check truth of clause j is coherent on the entire graph).

We write $\text{valid}(\mu)$ when the capped multiset μ represents a locally valid neighborhood according to the above conditions.

The modified CA local rule f is almost identical as the one from Theorem 9 and defined as follows (denoting again s the unique state such that $\mu(s, \epsilon) \geq 1$):

$$f(\sigma, \mu) = \begin{cases} e_{1-i} & \text{if } s = e_i \\ e_0 & \text{otherwise, and if } \neg \text{valid}(\mu), \\ e_0 & \text{otherwise, and if } s \text{ has type } (j, 0)\text{-truth-check and } (\sigma, \mu) \not\equiv_{loc} C_j, \\ s & \text{otherwise, and if } s \text{ is of type 1,} \\ \pi(s) & \text{otherwise.} \end{cases}$$

The key aspect of this new construction is that the correctness of the distributed truth check implemented in the CA above by states of type (j, m) -truth check rely on a modification of the considered FO formula. Let $\text{preimg}_j(y)$ be a FO formula expressing that the number of pre-images of y is either $> p_d$ or a multiple of p_j . We use the following modified definition of formula $\text{truth}(y)$:

$$\text{truth}(y) \stackrel{\text{def}}{=} \bigvee_{1 \leq j \leq d} \exists y^j : y^j \rightarrow y \wedge (\forall y' : y' \rightarrow y^j \Rightarrow \text{preimg}_j(y'))$$

which intuitively makes sense when y represents a configuration everywhere of type $\lambda(n)$, so y^j represents of configuration of type $(j, 0)$ -truth check everywhere. We this new definition of $\text{truth}(y)$, we introduce a strong dependence of the FO-formula on the matrix part of Ψ (by the presence of d for instance), which was not the case in Theorem 9.

The final FO formula ϕ is defined exactly as in Theorem 9 but using this modified version of $\text{truth}(y)$:

$$\phi \stackrel{\text{def}}{=} \mathbf{Q}'_1 y^1 \in \text{good}_1(y^1), \mathbf{Q}'_2 y^2 \in \text{good}_2(y^1, y^2), \dots, \mathbf{Q}'_n y^n \in \text{good}_n(y^1, \dots, y^n) : \text{truth}(y^n)$$

where the FO quantifier \mathbf{Q}'_i is \exists if the MSO quantifier \mathbf{Q}_i is existential, and \forall if \mathbf{Q}_i is universal.

The proof of Theorem 12 is based on two lemmas adapted from Lemma 10 and 11, and can be copied word for word from the proof of Theorem 9, but simply removing the connectedness hypothesis.

4 Consequences on FO model checking for CA

A set of (Σ, Δ) -labeled graphs, or graph language, is *MSO-definable* if it is of the form $\mathcal{G}(\Psi)$ for some MSO formula Ψ . It is *FOCA-definable* if it is of the form $\mathcal{G}(\phi, f)$ for some FO formula ϕ and some CA local rule f . From Theorem 12, we get the following immediate corollary.

► **Corollary 13.** *MSO-definable and FOCA-definable graph languages are the same.*

Since the translations given in Section 3 are effective, we also obtain equivalence of model checking problems. Using [33], this gives a characterization of decidability of FO model checking for CA orbits on f.g. groups.

► **Corollary 14.** *On any fixed graph, FO model checking for CA is many-one equivalent to MSO model checking. In particular, FO-model checking for CA on a f.g. group Γ is decidable if and only if Γ is virtually free.*

We can get a more precise result on graphs of bounded degree, but we need an additional lemma in order to apply Theorem 9. It is well-known that undecidability in MSO can be obtained using the MSO-definability of grids and encoding Turing-computations on it [17]. Moreover, MSO on bounded degree graphs actually allows to code quantification on edge sets and not only vertex sets [16], so that we can express the grid minor relation (and not only the fact that a fixed graph is a minor). The following lemma doesn't use any new idea, but it ensures the fact that all this encoding process can be done within a fixed prefix signature. It also gives a variant of the construction for Σ_1^1 -hardness (see [41]) using infinite grids and the recurring domino problem [25].

A $n \times n$ -grid (or simply a grid when n is not specified), is the directed finite graph with vertices $\{(i, j) : 1 \leq i, j \leq n\}$ and edges the set of pairs $((i, j), (i + 1, j))$ (the east edges) for $1 \leq j \leq n$ and $1 \leq i < n$, and pairs $((i, j), (i, j + 1))$ (the north edges) for $1 \leq j < n$ and $1 \leq i \leq n$. A ∞ -grid is the infinite directed graph with vertices $\mathbb{N} \times \mathbb{N}$ and same north/east adjacency relation.

► **Lemma 15.** *Fix some D . There exists a fixed quantifier prefix signature ρ for MSO such that, for any graph G of degree at most D that contains arbitrarily large grid as minors, deciding whether a given MSO formula in prenex form with prefix signature ρ is satisfied on G is undecidable. There also exists a fixed quantifier prefix signature ρ' for MSO such that, for any graph G of degree at most D that contains a ∞ -grid as minor, deciding whether a given MSO formula in prenex form with prefix signature ρ' is satisfied on G is Σ_1^1 -hard.*

This lemma together with Theorem 9 gives undecidability of model checking for a fixed FO formula expressed in the following corollary. For the first part of the corollary, we use the grid minor theorem [40] to translate the statement of the lemma in terms of treewidth [39],

and for the second part, it is known that the Cayley graph of a f.g. group which is not virtually free has a thick end (see [6]), hence it contains a ∞ -grid as a minor by Halin's grid theorem [24]. The decidable part of the Corollary comes from [14, 36].

► **Corollary 16.** *Fix some D . There is a fixed FO formula ϕ such that for any connected graph G of degree at most D , the set $\mathcal{CA}(G, \phi)$ is computable if and only if G has finite treewidth. Moreover, there is a FO formula ϕ' such that for any Cayley graph G of a f.g. group with at most D generators, the set $\mathcal{CA}(G, \phi)$ is:*

- *computable if the group is virtually free,*
- *Σ_1^1 -hard otherwise.*

In the context of modeling, it makes sense to consider distributed dynamical systems over arbitrary finite graphs. For instance automata networks (which are non-uniform CA on arbitrary finite graphs) are a well-established model for its use in the study of gene regulation networks [43, 31]. The theory of automata networks has largely grown around FO properties of orbits (typically fixed points) and their crucial dependence on the graph [38]. However, although many results deal with computational complexity in automata networks [11, 21], no natural undecidability result appeared so far to our knowledge. By our translation result from MSO, we can import Trakhtenbrot's theorem [19] to obtain an undecidability result for FO properties of CA orbits on finite graphs. It can intuitively be formulated as follows in the context of modeling: it is undecidable to know whether there is some finite interaction graph on which a given local interaction law (CA) induces a given dynamical property (FO). By the way, this corollary doesn't need Trakhtenbrot's theorem since we have all the expressive power of MSO (not only FO logic on graphs), and we can obtain it for a fixed FO formula. For instance, it follows directly from Theorem 9 and the techniques of Lemma 15 (see also [17, Theorem 5.6]).

► **Corollary 17.** *There exists a FO formula ϕ such that the following problem is undecidable: given some input CA local rule f , decide whether there exists a finite graph G with $F_{G,f} \models \phi$.*

5 Cayley graphs and domino problems

The case of Cayley graphs of f.g. groups is particular for our approach in two relevant ways. First, we can express in MSO that a set of vertices is infinite.

► **Lemma 18.** *For any $D \geq 1$, there is an MSO formula $\Psi(X)$ such that, on any Cayley graph G of some f.g. group with D generators and any assignment α , it holds that $(G, \alpha) \models \Psi(X)$ if and only if $\alpha(X)$ is infinite.*

From this lemma, it makes sense to extend the signature of FO logic with the addition of new relation $\overset{\infty}{\cong}$ on configurations which is at the heart of the 'Garden of Eden' theorem [12]: we write $c \overset{\infty}{\cong} c'$ whenever $\{v : c_v \neq c'_v\}$ is finite. We denote by $\text{FO}(\overset{\infty}{\cong})$ the extension of FO signature by adding relation $\overset{\infty}{\cong}$. By Lemma 18, this extension remains within MSO. Precisely, in any fixed Cayley graph of a f.g. group and by a straightforward extension of Theorem 8, we can compute from any formula in $\text{FO}(\overset{\infty}{\cong})$ and CA local rule, an equivalent MSO formula. We deduce that $\text{FO}(\overset{\infty}{\cong})$ model checking for CA is decidable on some f.g. group exactly when MSO model checking is, and exactly when FO model checking for CA is.

Besides, if Γ is a f.g. group and G_1 and G_2 two Cayley graphs of Γ with two different sets of generators Δ_1 and Δ_2 , then the definable CA global maps $F : S^\Gamma \rightarrow S^\Gamma$ are the same on G_1 and G_2 . More precisely, there is a computable translation τ on CA local maps such that for any local map f for G_1 , it holds: $F_{G_1,f} = F_{G_2,\tau(f)}$. This simply comes from the fact that we can translate Δ_1 -walks into equivalent Δ_2 -walks.

Therefore, if ϕ is a FO formula, we have that the sets $\mathcal{CA}(\phi, G_1)$ and $\mathcal{CA}(\phi, G_2)$ are actually Turing-equivalent. Said differently, by Theorem 8, any FO formula (actually any $\text{FO}(\overset{\infty}{=})$ formula) defines a fragment of MSO logic whose model checking problem's Turing degree is independent of the choice of generators on a f.g. group Γ . It turns out that such fragments naturally capture the domino problem and its classical variants.

Given some finite set S , a domino specification \mathcal{D} is a set of pairs $D_\delta \subseteq S^2$ for each $\delta \in \Delta$. A configuration $c \in S^V$ is said \mathcal{D} -valid for some graph if for any $v, v' \in V$ it holds: $(v, v') \in E_\delta$ implies $(c_v, c_{v'}) \in R_\delta$. The domino problem on a fixed graph, consists in deciding given \mathcal{D} whether there exists a \mathcal{D} -valid configuration. The seeded domino problem consists in deciding given \mathcal{D} and $s_0 \in S$ whether there exists a \mathcal{D} -valid configuration where s_0 occurs at some vertex. Finally, the recurring domino problem consists in deciding given \mathcal{D} and $s_0 \in S$ whether there exists a \mathcal{D} -valid configuration where s_0 occurs infinitely often.

► **Theorem 19.** *Fix any Cayley graph G of any f.g. group, then:*

- *domino problem* $\equiv_T \exists x, x \rightarrow x$,
- *seeded domino problem* $\equiv_T \exists x, \exists y, x \rightarrow x \wedge y \rightarrow x \wedge x \neq y$,
- *recurring domino problem* $\equiv_T \exists x, \exists y, x \rightarrow x \wedge y \rightarrow x \wedge \neg(x \overset{\infty}{=} y)$,

where $\equiv_T \phi'$ means Turing-equivalent to the set $\mathcal{CA}(\phi, G)$ (model checking of ϕ for CAs on G).

The recurring domino problem is Σ_1^1 -hard on \mathbb{Z}^2 [25], as well as the model checking of the corresponding $\text{FO}(\overset{\infty}{=})$ formula from Theorem 19. It is just an existential formula, but it crucially uses relation $\overset{\infty}{=}$. We can actually also obtain Σ_1^1 -hardness on \mathbb{Z}^2 with a pure FO formula with just one quantifier alternation, using pre-image counting trickery to check finiteness of a set and a reduction from the recurring domino problem.

► **Theorem 20.** *The problem $\mathcal{CA}(\phi, \mathbb{Z}^2)$ is Σ_1^1 -hard where ϕ is the following formula:*

$$\phi \stackrel{\text{def}}{=} \exists y, y \rightarrow y \wedge \forall y', \forall y_1, \forall y_2, \forall y_3, (y' \neq y \wedge y' \rightarrow y \wedge \bigwedge_{i \neq j} y_i \rightarrow y') \Rightarrow \bigvee_{i \neq j} y_i = y_j.$$

6 Perspectives

We see several interesting research directions inspired by the approach taken in this work.

First, we believe that the dependence of ϕ on the degree or the number of generators in Corollary 16 is an artifact that can be removed with more work in the proof of Lemma 15. The same proof techniques should also provide hardness result at any level of the analytical hierarchy.

Then, this corollary should be put into perspective with the Ballier-Stein conjecture [6] saying that the domino problem on a f.g. group is decidable if and only if the group is virtually free. On one hand, it seems natural to ask whether the recurring domino problem (or its equivalent FO formula from Theorem 19) can play the role of formula ϕ' in Corollary 16. On the other hand, N. Pytheas Fogg pointed us simple examples of 4-regular graphs having an ∞ -grid as subgraph on which the domino problem is decidable. So formula ϕ in Corollary 16 cannot be the FO formula expressing the existence of a fixed point (Turing-equivalent to the domino problem), and we wonder how simple such formula ϕ can be. Actually, we can ask a similar question for Corollary 17.

In general, we believe that the Turing degrees of FO-model checking problems for various concrete formulas is worth being investigated. As mentioned above, the Turing degree of all such model checking problems for a fixed FO formula is independent of the choice of

generators on f.g. groups, and we wonder how they change when changing the group among non virtually free groups. Injectivity of CA is a natural candidate that received little attention to our knowledge since the seminal result on \mathbb{Z}^2 [29].

Finally, we believe that there exists a fixed CA rule f for which the FO-model checking problem is undecidable on graph \mathbb{Z}^2 (the rule is fixed, the formula is given as input). While we see the proof ingredient to obtain this specifically for \mathbb{Z}^2 , we have no idea of whether it is always the case that undecidability of FO model checking for CA orbits can be obtained for a fixed CA rule on any f.g. which is not virtually free.

References

- 1 S. Amoroso and Y.N. Patt. Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures. *Journal of Computer and System Sciences*, 6(5):448–464, October 1972. doi:10.1016/s0022-0000(72)80013-8.
- 2 Pablo Arrighi and Gilles Dowek. Causal graph dynamics. *Information and Computation*, 223:78–93, February 2013. doi:10.1016/j.ic.2012.10.019.
- 3 Pablo Arrighi, Simon Martiel, and Vincent Nesme. Cellular automata over generalized cayley graphs. *Mathematical Structures in Computer Science*, 28(3):340–383, May 2017. doi:10.1017/s0960129517000044.
- 4 Nathalie Aubrun, Sebastián Barbieri, and Emmanuel Jeandel. *About the Domino Problem for Subshifts on Groups*, pages 331–389. Springer International Publishing, 2018. doi:10.1007/978-3-319-69152-7_9.
- 5 Nathalie Aubrun, Sebastián Barbieri, and Etienne Moutot. The Domino Problem is Undecidable on Surface Groups. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 46:1–46:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2019.46.
- 6 Alexis Ballier and Maya Stein. The domino problem on groups of polynomial growth. *Groups, Geometry, and Dynamics*, 12(1):93–105, March 2018. doi:10.4171/ggd/439.
- 7 Laurent Bartholdi. The domino problem for hyperbolic groups, 2023. arXiv:2305.06952.
- 8 Laurent Bartholdi and Ville Salo. Simulations and the lamplighter group. *Groups, Geometry, and Dynamics*, 16(4):1461–1514, November 2022. doi:10.4171/ggd/692.
- 9 R. Berger. The undecidability of the domino problem. *Mem. Amer. Math Soc.*, 66, 1966.
- 10 Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for your Mathematical Plays*, volume 2. Academic Press, 1982. chapter 25.
- 11 Florian Bridoux, Amélia Durbec, Kevin Perrot, and Adrien Richard. Complexity of fixed point counting problems in boolean networks. *Journal of Computer and System Sciences*, 126:138–164, June 2022. doi:10.1016/j.jcss.2022.01.004.
- 12 T. Ceccherini-Silberstein and M. Coornaert. *Cellular automata and groups*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2010. doi:10.1007/978-3-642-14034-1.
- 13 Bastien Chopard and Michel Droz. *Cellular Automata Modeling of Physical Systems*. Cambridge University Press, December 1998. doi:10.1017/cbo9780511549755.
- 14 Bruno Courcelle. The monadic second-order logic of graphs, ii: Infinite graphs of bounded width. *Mathematical Systems Theory*, 21(1):187–221, December 1988. doi:10.1007/bf02088013.
- 15 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, March 1990. doi:10.1016/0890-5401(90)90043-h.
- 16 Bruno Courcelle. The monadic second order logic of graphs vi: on several representations of graphs by relational structures. *Discrete Applied Mathematics*, 54(2-3):117–149, October 1994. doi:10.1016/0166-218x(94)90019-1.

- 17 Bruno Courcelle and Joost Engelfriet. *Graph structure and monadic second-order logic. A language-theoretic approach*. Encyclopedia of Mathematics and its applications, Vol. 138. Cambridge University Press, June 2012. Collection Encyclopedia of Mathematics and Applications, Vol. 138.
- 18 Alberto Dennunzio, Enrico Formenti, and Julien Provillard. Non-uniform cellular automata: Classes, dynamics, and decidability. *Information and Computation*, 215:32–46, June 2012. doi:10.1016/j.ic.2012.02.008.
- 19 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer Berlin Heidelberg, 1995. doi:10.1007/978-3-662-03182-7.
- 20 Louis Esperet, Ugo Giocanti, and Clément Legrand-Duchesne. The structure of quasi-transitive graphs avoiding a minor with applications to the domino problem, 2023. doi:10.48550/arXiv.2304.01823.
- 21 Guilhem Gamard, Pierre Guillon, Kevin Perrot, and Guillaume Theyssier. Rice-Like Theorems for Automata Networks. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, volume 187 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2021.32.
- 22 Martin Gardner. Mathematical games. *Scientific American*, 223(4):120–123, October 1970. doi:10.1038/scientificamerican1070-120.
- 23 Walter Gottschalk. *Some general dynamical notions*, pages 120–125. Springer Berlin Heidelberg, 1973. doi:10.1007/bfb0061728.
- 24 R. Halin. Über unendliche wege in graphen. *Mathematische Annalen*, 157(2):125–137, April 1964. doi:10.1007/bf01362670.
- 25 David Harel. *Recurring Dominoes: Making the Highly Undecidable Highly Understandable*, pages 51–71. Elsevier, 1985. doi:10.1016/s0304-0208(08)73075-5.
- 26 G. A. Hedlund. Endomorphisms and Automorphisms of the Shift Dynamical Systems. *Mathematical Systems Theory*, 3(4):320–375, 1969.
- 27 Benjamin Hellouin de Menibus, Victor H. Lutfalla, and Camille Noûs. *The Domino Problem Is Undecidable on Every Rhombus Subshift*, pages 100–112. Springer Nature Switzerland, 2023. doi:10.1007/978-3-031-33264-7_9.
- 28 Michael Hochman and Tom Meyerovitch. A characterization of the entropies of multi-dimensional shifts of finite type. *Annals of Mathematics*, 171(3):2011–2038, 2010. doi:10.4007/annals.2010.171.2011.
- 29 Jarkko Kari. Reversibility and surjectivity problems of cellular automata. *Journal of Computer and System Sciences*, 48(1):149–182, February 1994. doi:10.1016/s0022-0000(05)80025-x.
- 30 Jarkko Kari. Rice’s theorem for the limit sets of cellular automata. *Theoretical Computer Science*, 127(2):229–254, May 1994. doi:10.1016/0304-3975(94)90041-8.
- 31 S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, March 1969. doi:10.1016/0022-5193(69)90015-0.
- 32 Stephan Kreutzer and Siamak Tazari. On brambles, grid-like minors, and parameterized intractability of monadic second-order logic. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 354–364. SIAM, 2010. doi:10.1137/1.9781611973075.30.
- 33 Dietrich Kuske and Markus Lohrey. Logical aspects of cayley-graphs: the group case. *Annals of Pure and Applied Logic*, 131(1-3):263–286, January 2005. doi:10.1016/j.apal.2004.06.002.
- 34 Douglas Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, December 2020. doi:10.1017/9781108899727.
- 35 Marston Morse and G. A. Hedlund. Symbolic dynamics. *Amer. J. Math.*, 3:286–303, 1936.
- 36 David E. Muller and Paul E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985. doi:10.1016/0304-3975(85)90087-8.

- 37 Fabian Reiter. Distributed graph automata. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*. IEEE, July 2015. doi:10.1109/lics.2015.27.
- 38 Adrien Richard. Fixed points and connections between positive and negative cycles in boolean networks. *Discrete Applied Mathematics*, 243:1–10, July 2018. doi:10.1016/j.dam.2017.12.037.
- 39 Neil Robertson and P.D Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, September 1986. doi:10.1016/0196-6774(86)90023-4.
- 40 Neil Robertson and P.D Seymour. Graph minors. v. excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, August 1986. doi:10.1016/0095-8956(86)90030-4.
- 41 H. Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press, 1987.
- 42 Thomas Schwentick and Klaus Barthelmann. *Local normal forms for first-order logic with applications to games and automata*, pages 444–454. Springer Berlin Heidelberg, 1998. doi:10.1007/bfb0028580.
- 43 René Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563–585, December 1973. doi:10.1016/0022-5193(73)90247-6.
- 44 Wolfgang Thomas. *On logics, tilings, and automata*, pages 441–454. Springer Berlin Heidelberg, 1991. doi:10.1007/3-540-54233-7_154.