Testing Dynamic Environments: Back to Basics

Yonatan Nakar
Tel Aviv University, Israel

Dana Ron
Tel Aviv University, Israel

Abstract

We continue the line of work initiated by Goldreich and Ron (*Journal of the ACM, 2017*) on testing dynamic environments and propose to pursue a systematic study of the complexity of testing basic dynamic environments and local rules. As a first step, in this work we focus on dynamic environments that correspond to elementary cellular automata that evolve according to threshold rules.

Our main result is the identification of a set of conditions on local rules, and a meta-algorithm that tests evolution according to local rules that satisfy the conditions. The meta-algorithm has query complexity $\text{poly}(1/\epsilon)$, is non-adaptive and has one-sided error. We show that all the threshold rules satisfy the set of conditions, and therefore are $\text{poly}(1/\epsilon)$-testable. We believe that this is a rich area of research and suggest a variety of open problems and natural research directions that may extend and expand our results.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Property Testing

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.98

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2105.00759

1 Introduction

Property testing [12, 5] is the study of algorithms that distinguish between objects that have a given property and those that are far from having the property, by performing a small number of queries to the object. Goldreich and Ron [6] initiated the study of testing *dynamic environments*, which introduces a temporal aspect to property testing. In this context, the entity being tested changes with time, and is referred to as an *environment.*

Starting from some initial *configuration* (say, a vector or a matrix), the environment is supposed to evolve according to a prespecified *local* rule. The rule is local in the sense that the value associated with each location in the environment at time t is determined by the values of nearby locations at time $t-1$. The goal of the testing algorithm is then to distinguish between the case that the environment indeed evolves according to the rule, and the case in which the evolution significantly strays from obeying the rule. To this end, the algorithm can query the value of any location of the environment at any of the available time steps, as long as it does not “go back in time”. Namely, the algorithm cannot choose to query a location at time t after it has queried some location at time $t' > t$. We refer to this as the *time-conforming* requirement. The aim is to design time-conforming algorithms with low query complexity.

Goldreich and Ron [6] investigate the complexity landscape of testing dynamic environments from multiple angles. From a hardness perspective, they show that there are dynamic environments whose testing requires high query complexity and running time, and that adaptivity and time-conformity are relevant constraints which can significantly impact the query complexity. However, as we discuss in Section 1.4, relatively little is known regarding positive results for testing specific rules.

48th International Colloquium on Automata, Languages, and Programming (ICALP 2021).
Editors: Nikhil Bansal, Emanuela Merelli, and James Worrell; Article No. 98; pp. 98:1–98:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In our quest for understanding which natural families of dynamic environments can be tested efficiently, we propose to first “go back to the basics” and study testing in the simplest of dynamic environments. Namely, in this work we consider environments defined by one-dimensional configurations, which evolve according to local rules that are functions of the current location and its two immediate neighbors. These dynamic environments, originally introduced by von Neumann [13], have been extensively studied under the name of *Elementary Cellular Automata* [14] (see definition in Section 1.1). While these environments can be described in simple terms, they are nevertheless able to capture complex behavior.[1] Cellular automata have played a role in various research fields and applications. Examples include modeling physical [2] and chemical [7] systems, VLSI design [3], music generation [1], analyzing plant population dynamics [15], forest fire spread [17], city traffic [11], urban sprawl [8], and more.

As we discuss in Section 1.4, there are several hardness results (both regarding the query complexity and the running time) for testing dynamic environments that correspond to one-dimensional cellular automata (over non-binary alphabets) [6]. Hence, in order to obtain efficient algorithms, it is necessary to restrict the rules considered. In the current work, our main focus is on perhaps the most basic and natural rules, defined by threshold functions. Such functions have received much attention within the study of propagation of information/influence in networks (see, e.g., the review paper of Peleg [10], and the recent Ph.D. thesis of Zehmakan [16] and references within).

Our testers are based on a general meta-algorithm which works for rules that satisfy a set of conditions that we define. In essence, the conditions capture a certain type of behavior leading to ultimate convergence. This behavior induces a global structure on the environment which we exploit in our meta-algorithm.

We hope this work can serve as a basis for further extensions and generalizations, some of which we discuss shortly in Section 1.5.

1.1 Testing basic evolution rules

We now formally define the problems we study. We use $[m]$ to denote the set $\{0, ..., m-1\}$. For two integers n and m, let $\texttt{ENV} : [m] \times \mathbb{Z}_n \to \{0,1\}$ denote the evolving environment, and for any $t \in [m]$ let $\texttt{ENV}_t : \mathbb{Z}_n \to \{0,1\}$ (the environment at time t) be defined by $\texttt{ENV}_t(i) = \texttt{ENV}(t,i)$. In general, we refer to a function $\sigma : \mathbb{Z}_n \to \{0,1\}$ as a *configuration*. When convenient, we may view σ as a (cyclic) binary string of length n.

For a function (evolution rule) $\rho : \{0,1\}^3 \to \{0,1\}$, we say that $\texttt{ENV}$ *evolves according to* ρ, if for every $i \in \mathbb{Z}_n$ and $t > 0$, we have that $\texttt{ENV}_t(i) = \rho(\texttt{ENV}_{t-1}(i-1), \texttt{ENV}_{t-1}(i), \texttt{ENV}_{t-1}(i+1))$, where all operations are modulo n. We use $\mathcal{E}^{\rho}_{m,n}$ to denote the set of environments $\texttt{ENV} : [m] \times \mathbb{Z}_n \to \{0,1\}$ that evolve according to ρ. As in [6], we employ the standard notion of distance used in property testing and say that $\texttt{ENV} : [m] \times \mathbb{Z}_n \to \{0,1\}$ is *ϵ-far from evolving according to ρ* (ϵ-far from $\mathcal{E}^{\rho}_{m,n}$) if $|\{(t,i) : \texttt{ENV}(t,i) \neq \texttt{ENV}'(t,i)\}| > \epsilon mn$ for every $\texttt{ENV}' \in \mathcal{E}^{\rho}_{m,n}$.[2]

[1] Some rules are even Turing complete [4].

[2] In the context of dynamic environments, this notion of distance can be interpreted as capturing “measurement errors” due to some noise process. Namely, it can be viewed as allowing the testing algorithm to accept not only “perfect” environments, but also environments that correspond to a correct evolution with a bounded fraction of corruptions. Also note that being ϵ-far from evolving according to ρ does not simply translate to there being an ϵ-fraction of pairs (t,i) for which $\texttt{ENV}_t(i) \neq \rho(\texttt{ENV}_{t-1}(i-1), \texttt{ENV}_{t-1}(i), \texttt{ENV}_{t-1}(i+1))$ (which would be trivial to test).

Given n, m, and a distance parameter $\epsilon \in (0,1)$, a testing algorithm for evolution according to a rule ρ should distinguish with constant success probability between the case that an environment ENV belongs to $\mathcal{E}^{\rho}_{m,n}$ and the case that it is ϵ-far from $\mathcal{E}^{\rho}_{m,n}$. To this end, the algorithm is given query access to ENV, where a query on a pair (t,i) cannot follow any query on (t',i') for $t' > t$. We are interested in bounding both the total number of queries performed by the algorithm (as a function of ϵ, and possibly m and n) and the maximum number of queries it performs at any time step (which we refer to as its *temporal query complexity*).

1.2 Our results

We identify several conditions on local rules (which are formally stated in Section 3), such that if a rule ρ satisfies these conditions, then evolution according to ρ can be tested with query complexity poly$(1/\epsilon)$ with one-sided error. Our testers have the advantage that they are non-adaptive, and therefore, in particular, time-conforming.

▶ **Theorem 1.** *Let Ψ be the set of conditions specified in Section 3. For every rule ρ that satisfies the conditions in Ψ, it is possible to test evolution according to ρ by performing $O(1/\epsilon^4)$ queries. Furthermore, the testing algorithm is non-adaptive and has one-sided error.*

To establish Theorem 1, we present a *meta-algorithm* for testing evolution and prove its correctness for rules that satisfy the aforementioned conditions (the set Ψ). It is a meta-algorithm in the sense that it is based on certain subroutines that are rule-specific (but have a common functionality of detecting violations of evolution according to the tested rule). We provide a high-level discussion of the conditions and the algorithm in Section 1.3.

Our main application of the meta-algorithm is to the natural family of threshold rules.

▶ **Definition 1.** *We say that a rule $\rho : \{0,1\}^3 \to \{0,1\}$ is a* threshold *rule if there exist a threshold integer $0 \le b \le 3$ and a bit $\alpha \in \{0,1\}$ such that $\rho(\beta_1,\beta_2,\beta_3) = \alpha$ if and only if $\beta_1 + \beta_2 + \beta_3 \ge b$.*

We prove:

▶ **Theorem 2.** *For each threshold rule ρ, evolution according to ρ can be tested with query complexity $O(1/\epsilon^4)$. Furthermore, the testing algorithm is non-adaptive and has one-sided error.*

We also show that the conditions hold for two additional (non-threshold) rules, so the applicability of our meta-algorithm is more general (for details, see full version of this paper [9]). We believe that appropriate (perhaps more complex) variants of our algorithm can be used to test an even larger variety of basic local rules (see Section 1.5), where we conjecture that this is true for all rules that ultimately converge. Interestingly, while the two additional rules are not threshold rules as per Definition 1, they can be represented as weighted threshold rules (which are a subclass of ultimately converging rules).

1.3 The high-level ideas behind our results

In this high-level discussion, we assume for simplicity that $m \ge n$ (the case $m < n$ can be essentially reduced to this case).

1.3.1 Convergence, final/non-final locations and prediction functions

To give an intuition on the convergence behavior that our conditions capture, it is useful to first discuss the notion of *ultimate convergence.* A rule ρ ultimately converges if, for any initial configuration $\texttt{ENV}_0$, an environment evolving according to ρ converges after a bounded number of steps to either a single final configuration or to a constant number of configurations between which it alternates. For example, consider the majority rule (threshold 2). Unless the initial configuration is $(01)^{n/2}$, the environment ultimately converges to some configuration that consists of blocks of 0s and 1s of size at least 2 each (and if it is $(01)^{n/2}$, then it alternates between $(01)^{n/2}$ and $(10)^{n/2}$).

Once an environment converges, testing is straightforward since we can easily predict the values of locations in future time steps and then verify that indeed they hold the predicted values (or else we reject). The issue, however, is that convergence is not ensured to be reached after a small number of time steps.[3] In other words, knowing that a rule ultimately converges cannot be exploited directly. Hence, the challenge is to identify and formalize conditions that allow for "pre-convergence prediction". Namely, conditions that imply the ability to predict future values of locations based on the current values of these and other locations (before convergence is reached).

In this context, our conditions try to formalize the idea that rules exhibit a certain *local* convergence, which "expands" with time. The first ingredient of our approach is the observation that, in the case of the majority rule, if at any time step t, $\texttt{ENV}_t(i) \in \{\texttt{ENV}_t(i-1), \texttt{ENV}_t(i+1)\}$, then $\texttt{ENV}_{t'}(i) = \texttt{ENV}_t(i)$ for any $t' > t$ (operations are modulo n). We say in such a case that location i is *final* at time t (in $\texttt{ENV}$). Otherwise it is *non-final.* Crucially for us, whether a location i is final or not at a certain time step depends solely on its local neighborhood at that time (and can hence be verified with a constant number of queries).

An important property of a location being final at time t (in addition to converging to their final value, up to alternations), is the aforementioned expansion (or "transmission of finality"). Namely, a location i that is non-final at time t becomes final at time $t+1$ if either $i-1$ or $i+1$ is final at time t (possibly both). Furthermore, it cannot become final if both its neighbors are non-final. Another related property of final locations is that (under certain circumstances), they can be used to predict the values of locations that become final in the future, based on a (rule-specific) *prediction function.* A similar statement holds for non-final locations (though the circumstances are different).

1.3.2 The meta-algorithm: the grid and violating pairs

Based on these properties (which are formalized in the conditions we introduce), our (meta) algorithm works in two stages. In the first stage, it queries the environment at time $t_1 = \Theta(\epsilon m)$ on $O(1/\epsilon^2)$ equally spaced locations, which we refer to as *the grid locations*, and their local neighborhoods. This allows the algorithm to determine which of the grid locations are final at time t_1 and which are non-final. If the answers it gets are not consistent with any environment that evolves according to ρ (in which case we say that the grid is *not feasible*), then it rejects.

In its second stage, the algorithm uniformly samples $O(1/\epsilon)$ random time-location pairs (t, i) and queries $\texttt{ENV}_t$ on i and its local neighborhood. It then checks whether the answers are consistent with the answers to queries in the first stage (on the grid locations and their neighborhoods) or constitute a *violation.* The definition of consistency/violation is based on the aforementioned prediction functions of the tested rule.

[3] In fact, there are initial configurations that require $\Omega(n)$ steps before they ultimately converge.

One may have hoped that such a consistency check is sufficient, in the sense that all (or almost all) pairs (t, i) can be predicted based on the answers to the queried grid locations. Unfortunately, this is not the case. There are (possibly many) pairs (t, i) whose 0/1 values are not determined given the first-stage answers. However, we show that such pairs are constrained in a different way (in environments that evolve according to ρ): their location must have become final by time $t_2 = t_1 + \Delta$, where Δ is the distance between grid location. Hence, for each selected pair (t, i), the algorithm also queries $\texttt{ENV}_{t_2}$ on location i (and its neighborhood) and checks consistency with the queried locations at time t_2.

1.3.3 On the analysis of the algorithm and "backward prediction"

To show that the algorithm always accepts environments that evolve according to the tested rule ρ, we prove that our definition of violation is such that there are no violations in such environments (assuming ρ satisfies the aforementioned conditions). The more involved part of the analysis is proving that if the environment $\texttt{ENV}$ is ϵ-far from evolving according to ρ, then the algorithm will detect this with probability at least 2/3. To this end, we prove the contrapositive statement. Namely, we show that if the algorithm accepts with probability at least 2/3, then there exists an environment that evolves according to ρ and is ϵ-close to $\texttt{ENV}$. This is done by showing that we can construct an initial configuration $\texttt{ENV}'_0$, such that if we let it evolve according to ρ, resulting in an environment $\texttt{ENV}' \in \mathcal{E}^{\rho}_{m,n}$, then we can upper bound the number of pairs (t, i) such that $\texttt{ENV}_t(i) \neq \texttt{ENV}'_t(i)$ by ϵmn.

Here we build on a useful property of the prediction functions, by which they allow us a certain "prediction back in time". Namely (for t_1 and t_2 as mentioned above), we use the queried grid locations at time t_1 as well as some locations at time t_2 (which have not been queried) to determine values of locations at the earlier time 0 in $\texttt{ENV}'$. We prove that this can be done in a way that ensures that $\texttt{ENV}'$ agrees with $\texttt{ENV}$ on all pairs (t, i) that are not violating.

1.4 A short overview of the results in [6]

As stated earlier, the study of testing dynamic environments was initiated by Goldreich and Ron [6], who present several general results as well as analyze two natural specific rules. We first provide a short overview of their main general results.

They prove that the query complexity of testing (one-dimensional) rules may have high query complexity. Specifically, they show that there exists a constant $c > 0$ and an evolution rule $\rho : \Sigma^3 \to \Sigma$ such that any tester of evolution according to ρ requires $\Omega(n^c)$ queries.[4] They also prove that testing dynamic environments may be NP-Hard, provided that the temporal query complexity is "significantly sublinear" (where $f(x)$ is significantly sublinear if $f(x) < x^{1-\Omega(1)}$). More precisely, they show that for every constant $c > 0$ there exists an evolution rule $\rho : \Sigma^3 \to \Sigma$ such that no (time-conforming) polynomial-time testing algorithm with temporal query complexity n^{1-c} can test whether n-sized environments evolve according to ρ (assuming $\mathcal{NP} \not\subseteq \mathcal{BPP}$). Their general results also include a theorem concerning the usefulness of adaptivity in testing dynamic environments, a study of the relation between testing and learning dynamic environments, and a result on the power of being non time-conforming.

[4] Observe that it is possible to test the evolution according to any rule ρ over configurations of size n by performing $O(n + 1/\epsilon)$ queries (n queries to the initial configuration and $O(1/\epsilon)$ uniformly selected queries elsewhere). To get sublinear temporal query complexity, a total of $O(n/\epsilon)$ uniformly selected queries suffice (by applying a simple union bound over all possible initial configurations).

Goldreich and Ron [6] also provide testers for evolution according to two specific (classes of) rules. The first is the class of linear rules, which in the binary 1-dimensional case corresponds to the XOR rule in elementary cellular automata. They show that for any $d \geq 1$ and any field Σ of prime order, there exists a constant $\gamma < d$ such that the following holds. For any linear rule $\rho : \Sigma^{3^d} \rightarrow \Sigma$ there exists a time-conforming oracle machine of (total) time complexity $\text{poly}(1/\epsilon) \cdot n^{\gamma}$ that tests the consistency of an evolving environment with respect to ρ. Furthermore, the tester is non-adaptive and has one-sided error.

Their second specific positive result, loosely stated, captures fixed-speed movement of objects in one-dimension such that colliding objects stop forever. They present a (time-conforming) algorithm of (total) time complexity $\text{poly}(1/\epsilon)$ that tests the consistency of evolving environments with respect to that rule.

1.5 Future directions

Basic dynamic environments. A natural question that arises is whether a more nuanced version of the set of conditions formalized in this paper and the meta-algorithm can be defined and proved to work for other rules in the realm of basic dynamic environments. Indeed, preliminary results suggest that several other rules that ultimately converge exhibit behaviors that "resemble" the ones captured by our conditions. This leads us to the following conjecture.

▶ **Conjecture.** *If a rule ρ ultimately converges, then it is poly$(\frac{1}{\epsilon})$-testable.*

While our meta-algorithm does not apply to rules that do not ultimately converge, there are natural rules that fall under this category (the XOR rule for instance). This raises the question of whether $\text{poly}(1/\epsilon)$ testers exist for such rules. The answer is that there are $\text{poly}(1/\epsilon)$-testable rules that do not ultimately converge, but as we'll see, the question should be slightly rephrased.[5] To give one example, for the rule ρ defined as $\rho(x, y, z) = x$, each configuration is simply a copy of the previous configuration, shifted one location to the right. That is, while an environment evolving according to this rule does not, technically, ultimately converge, this rule is trivially $\text{poly}(1/\epsilon)$-testable. However, this particular rule and other rules that are capable of producing such "shifting behaviors" also have the property of not being *symmetric* (i.e., it does not hold that $\rho(x, y, z) = \rho(z, y, x)$ for every x, y, z). Hence, one way to rephrase the question is restricting it to symmetric rules.

▶ **Open Problem 1.** *Are there any symmetric rules that do not ultimately converge and are poly$(1/\epsilon)$-testable?*

Another way to rephrase this question is to define a more general notion of ultimate convergence. Specifically, we say that a rule ρ *ultimately converges up to a shift* if, for any initial configuration $\texttt{ENV}_0$, an environment evolving according to ρ converges after a bounded number of steps to a constant number of configuration *equivalence classes* between which it alternates, where two configurations are *equivalent* if they are equal *up to a shift.*

▶ **Open Problem 2.** *Are there any non-symmetric rules that do not ultimately converge up to a shift and are poly$(1/\epsilon)$-testable?*

As mentioned in Section 1.4, it has been shown in [6] that the XOR rule is sublinearly testable. However, the query complexity of the tester depends on the size of the environment and is only mildly sublinear (the complexity is $O(n^{0.8})$ for an environment of size n). This

[5] We thank one of the anonymous reviewers of this paper for pointing this out.

raises the question of whether there exists a tester for the XOR rule with significantly lower query complexity (maybe even polylogarithmic). Another question that can be raised is whether there are other symmetric rules, ones that do not ultimately converge, that can be tested with a sublinear query complexity that depends on the size of the environment.

▶ **Open Problem 3.** *Which symmetric rules that do not ultimately converge can be tested with query complexity that is sublinear in (but strictly grows with) the size of the environment?*

More general dynamic environments. Building on the ideas for testing basic dynamic environments, it may be possible to venture into more general environments. One such generalization is to consider rules that depend on more than just the three locations constituting the immediate neighborhood. Other generalizations are to environments and rules over non-binary values, higher dimensions, and environments that evolve on more general graphs.

Non-deterministic rules. We also suggest considering local rules that are non-deterministic in the sense that given some configuration, the rule allows several configurations to follow. An example of one such rule, which can be thought of as a relaxation of the OR rule, is the rule in which each value is restricted to be monotonically non-decreasing with respect to the previous values at the location's neighborhood.

Missing details

Due to space constraints, not all details appear in this extended abstract, and can be found in the full version of this paper [9].

2 Preliminaries

In addition to the basic definitions provided in Section 1.1 regarding testing dynamic environments, here we introduce several more definitions and notations.

In all that follows, when performing operations on locations $i \in \mathbb{Z}_n$, these operations are modulo n. For a pair of locations $i, j \in \mathbb{Z}_n$ we use $[i, j]$ to denote the sequence $i, i+1, \ldots, j$ (so that it is possible that $j < i$).

▶ **Definition 2.** *For a location $i \in \mathbb{Z}_n$ and an integer r, the r-neighborhood of i, denoted $\Gamma_r(i)$, is the sequence $[i-r, i+r]$. For a set of locations $I \subseteq \mathbb{Z}_n$, we let $\Gamma_r(I)$ denote the set of locations in the union of sequences $[i-r, i+r]$ taken over all $i \in I$.*

▶ **Definition 3.** *For an integer n and a local rule ρ, let $\mathcal{M}_\rho(n)$ denote the (deterministic) state machine that is defined as following. Each state of $\mathcal{M}_\rho(n)$ corresponds to a different configuration $\sigma : \mathbb{Z}_n \to \{0, 1\}$. If a state corresponds to a configuration σ, then it has a single transition going to the state corresponding to the configuration that results from applying ρ to σ.*

The period of $\mathcal{M}_\rho(n)$, denoted $p_\rho(n)$, is the longest size of a (directed) cycle in $\mathcal{M}_\rho(n)$. If there exists a constant p such that $p_\rho(n) = p$ ($p_\rho(n) \leq p$) for every sufficiently large n, then we say that ρ has period (at most) p, and that ρ ultimately converges.

Observe that for every $\mathcal{M}_\rho(n)$, each strongly connected component in $\mathcal{M}_\rho(n)$ is either a single state with no edges in the component or a cycle (where in particular, the cycle may be a self-loop). For example, if ρ is the OR function, then it has period 1 (as it contains only two cycles: one is a self-loop for the state corresponding to the configuration 1^n and the other is a self loop for the state corresponding to the configuration 0^n). On the other hand, there are rules, such as XOR, for which $p_\rho(n) = \Omega(n)$.

▶ **Definition 4.** *For two locations $i, i' \in \mathbb{Z}_n$, we let $\overrightarrow{\text{dist}}(i,i') = i' - i$ denote the directed distance from i to i, and let $\text{dist}(i,i') = \min\{\overrightarrow{\text{dist}}(i,i'), \overrightarrow{\text{dist}}(i',i)\}$ denote the (undirected) distance.*

Note that since operations on locations are modulo n, we have that $\overrightarrow{\text{dist}}(i,i') \le n-1$, while $\text{dist}(i,i') \le n/2$ for all $i, i' \in \mathbb{Z}_n$.

▶ **Definition 5.** *For $t \in [m]$ and $i \in \mathbb{Z}_n$, we refer to (t,i) as a* time-location pair *(or simply* pair*).*

Given two locations, $i, i' \in \mathbb{Z}_n$ and two time steps $t, t' \in [m]$ where $t > t'$, we say that the pair (t,i) descends *from the pair (t',i') if $\text{dist}(i,i') \le t - t'$. We say that (t,i) is a* descendant *of (t',i') and that (t',i') is an* ancestor *of (t,i).*

▶ **Definition 6.** *For an integer r, an r-*pattern *is a string in $\{0,1\}^r$.*

3 The Conditions

Let $\rho : \{0,1\}^3 \to \{0,1\}$ be a local rule. We present several conditions, such that if they all hold, then the rule ρ can be tested with $\text{poly}(1/\epsilon)$ queries. These conditions capture properties of local rules that can be exploited by our (meta) algorithm.

The conditions are defined with respect to a constant (integer) k (which depends on ρ, but for the sake of simplicity we suppress the dependence on ρ and use k rather than k_ρ), and a partition of all $(2k+1)$-patterns.[6] The partition is denoted by $(\mathcal{F}_\rho, \overline{\mathcal{F}}_\rho)$, where $\mathcal{F}$ stands for *final* and $\overline{\mathcal{F}}$ for *non-final*.

We shall say that a pair (t,i) is final (respectively, non-final) *with respect to* $\texttt{ENV}$ *and* ρ if $\texttt{ENV}_t(\Gamma_k(i)) \in \mathcal{F}_\rho$ (respectively, $\overline{\mathcal{F}}_\rho$). Roughly speaking, if (t,i) is final (with respect to $\texttt{ENV}$ and ρ), then location i does not change from time t and onward (or, more generally, $\texttt{ENV}_{t'}(i)$ for $t' > t$ can be predicted based on $\texttt{ENV}_t(i)$). Furthermore, if (t,i) is non-final, then $(t+1,i)$ is final if and only if $(t, i-1)$ or $(t-1, i+1)$ is final (so that finality is "infectious").

In our statements of the conditions, we make use of the parity function, which we denote by $\mathsf{parity} : \mathbb{N} \to \{0,1\}$ (so that $\mathsf{parity}(x) = 1$ if x is odd and $\mathsf{parity}(x) = 0$ if x is even).

Before each of the conditions is stated formally, we give a short, informal description. It will also be useful to have a running example of a specific rule ρ, which is the majority rule. Namely, $\text{MAJ}(\beta_1, \beta_2, \beta_3) = 1$ for any three bits $\beta_1, \beta_2, \beta_3$, if and only if $\beta_1 + \beta_2 + \beta_3 \ge 2$. For the majority rule, $k = 1$, and $\mathcal{F}_{\text{MAJ}} = \{111, 110, 011, 000, 001, 100\}$ (so that $\overline{\mathcal{F}}_{\text{MAJ}} = \{101, 010\}$).

The first condition says that if a location is final, then it remains final.

▶ **Condition 1.** *Let $\texttt{ENV} \in \mathcal{E}^\rho_{m,n}$ be an environment that evolves according to ρ. For any time step $t \in [m-1]$ and location $i \in \mathbb{Z}_n$, if $\texttt{ENV}_t(\Gamma_k(i)) \in \mathcal{F}_\rho$, then $\texttt{ENV}_{t+1}(\Gamma_k(i)) \in \mathcal{F}_\rho$.*

Indeed, for the majority rule, if $\texttt{ENV}_t(\Gamma_1(i)) = 111$, then $\texttt{ENV}_{t+1}(\Gamma_1(i)) = 111 \in \mathcal{F}_{\text{MAJ}}$, if $\texttt{ENV}_t(\Gamma_1(i)) = 110$, then $\texttt{ENV}_{t+1}(\Gamma_1(i)) \in \{110, 111\} \subset \mathcal{F}_{\text{MAJ}}$, and if $\texttt{ENV}_t(\Gamma_1(i)) = 110$, then $\texttt{ENV}_{t+1}(\Gamma_1(i)) \in \{110, 111\} \subset \mathcal{F}_{\text{MAJ}}$ (analogous statements hold for $\texttt{ENV}_t(\Gamma_1(i)) \in \{000, 001, 100\}$).

[6] For the local rules we apply our conditions to, k is either 0 or 1, but using a variable parameter k will hopefully allow to extend our results more easily.

The second condition says that if a location is non-final, then it can become final in one time step if and only if it has a final neighbor.

▶ **Condition 2.** *Let* $\mathtt{ENV} \in \mathcal{E}^{\rho}_{m,n}$ *be an environment that evolves according to* ρ*. For any time step* $t \in [m-1]$ *and location* $i \in \mathbb{Z}_n$*, if* $\mathtt{ENV}_t(\Gamma_k(i)) \in \overline{\mathcal{F}}_\rho$*, then* $\mathtt{ENV}_{t+1}(\Gamma_k(i)) \in \mathcal{F}_\rho$ *if and only if* $\mathtt{ENV}_t(\Gamma_k(i-1)) \in \mathcal{F}_\rho$ *or* $\mathtt{ENV}_t(\Gamma_k(i+1)) \in \mathcal{F}_\rho$ *(or both).*

For the majority rule, consider the case that $\mathtt{ENV}_t(\Gamma_1(i)) = 101$ (so that it belongs to $\overline{\mathcal{F}}_{\text{MAJ}}$). In this case, $\mathtt{ENV}_t(\Gamma_1(i-1)) \in \{110, 010\}$ and $\mathtt{ENV}_t(\Gamma_1(i+1)) \in \{011, 010\}$. If $\mathtt{ENV}_t(\Gamma_1(i-1)) = 110$ (which belongs to $\mathcal{F}_{\text{MAJ}}$), then $\mathtt{ENV}_{t+1}(\Gamma_1(i)) \in \{110, 111\} \subset \mathcal{F}_{\text{MAJ}}$, and the case that $\mathtt{ENV}_t(\Gamma_1(i+1)) = 011$ is analogous. On the other hand, if both $\mathtt{ENV}_t(\Gamma_1(i-1)) = 010$ and $\mathtt{ENV}_t(\Gamma_1(i+1)) = 010$ (so that they both belong to $\overline{\mathcal{F}}_{\text{MAJ}}$), then $\mathtt{ENV}_{t+1}(\Gamma_1(i)) = 010$ (and it belongs to $\overline{\mathcal{F}}_{\text{MAJ}}$ as well). Note that, if for every location $i \in \mathbb{Z}_n$, it holds that $\mathtt{ENV}_0(\Gamma_1(i)) \in \{010, 101\}$ (that is, every location in the initial configuration is non-final), then no location would ever become final throughout the evolution of the rule. In particular, in this case the environment alternates between $(01)^{n/2}$ and $(10)^{n/2}$, where all the locations are non-final.

The first two conditions intuitively imply that one can determine whether certain locations are final or non-final using particular "past" locations that are known to be final or non-final. The next two conditions capture the idea that the actual *values* at certain locations (and not only whether or not they are final) can also be determined based on past locations.

In particular, the third condition captures how values at locations that are final at a certain time step can be predicted using a function that depends on "past" final locations from which they descend (and to which they are closest).

▶ **Condition 3.** *Let* $\mathtt{ENV} \in \mathcal{E}^{\rho}_{m,n}$ *be an environment that evolves according to* ρ*. There exists a function* $f_\rho : \{0,1\}^3 \to \{0,1\}$ *for which the following holds. First,* f_ρ *is the XOR of its first argument and a subset of the other two arguments. Second, let* (t, i) *and* (t', i') *be any two pairs such that* (t, i) *descends from* (t', i')*,* $\mathtt{ENV}_t(\Gamma_k(i)), \mathtt{ENV}_{t'}(\Gamma_k(i')) \in \mathcal{F}_\rho$*, and for every* $i'' \neq i'$ *satisfying* $\text{dist}(i, i'') \leq \text{dist}(i, i')$ *it holds that* $\mathtt{ENV}_{t'}(\Gamma_k(i'')) \in \overline{\mathcal{F}}_\rho$*. Then*

$$\mathtt{ENV}_t(i) = f_\rho(\mathtt{ENV}_{t'}(i'), \textsf{parity}(t - t'), \textsf{parity}(\text{dist}(i, i'))) .$$

For the majority rule, f_{MAJ} is simply the identity function on its first argument, namely, $f_{\text{MAJ}}(\beta, \cdot, \cdot) = \beta$.

The fourth condition captures how locations that are non-final at a certain time step can be predicted using a function that depends on "past" non-final locations from which they descend (conditioned on there not being any final location among its ancestors in that past time step).

▶ **Condition 4.** *Let* $\mathtt{ENV} \in \mathcal{E}^{\rho}_{m,n}$ *be an environment that evolves according to* ρ*. There exists a function* $h_\rho : \overline{\mathcal{F}}_\rho \times \{0,1\} \times \mathbb{Z}_n \to \overline{\mathcal{F}}_\rho$ *for which the following holds. First,* h_ρ *is reversible in the sense that for each fixed* $\tau \in \overline{\mathcal{F}}_\rho$*,* $\beta \in \{0,1\}$ *and* $\ell \in \mathbb{Z}_n$*, there exists a unique* τ' *such that* $h_\rho(\tau', \beta, \ell) = \tau$*. Second, let* (t, i) *and* (t', i') *be any two pairs such that* (t, i) *descends from* (t', i')*,* $\mathtt{ENV}_t(\Gamma_k(i)), \mathtt{ENV}_{t'}(\Gamma_k(i')) \in \overline{\mathcal{F}}_\rho$*, and* $\mathtt{ENV}_{t'}(\Gamma_k(i'')) \in \overline{\mathcal{F}}_\rho$ *for every* i'' *such that* (t, i) *descends from* (t', i'')*. Then*

$$\mathtt{ENV}_t(\Gamma_k(i)) = h_\rho(\mathtt{ENV}_{t'}(\Gamma_k(i')), \textsf{parity}(t - t'), \overrightarrow{\text{dist}}(i', i)) .$$

For the majority rule, $h_{\text{MAJ}}(010, \beta, x) = 010$ if $\beta \oplus \textsf{parity}(x) = 0$ and $h_{\text{MAJ}}(010, \beta, x) = 101$ if $\beta \oplus \textsf{parity}(x) = 0$. Similarly, $h_{\text{MAJ}}(101, \beta, x) = 101$ if $\beta \oplus \textsf{parity}(x) = 0$ and $h_{\text{MAJ}}(101, \beta, x) = 010$ if $\beta \oplus \textsf{parity}(x) = 1$.

The additional two conditions presented below are a bit more involved than Conditions 1–4, and perhaps initially less intuitive. They do not play a role in the definition of the meta algorithm, but are applied in the proof of Lemma 2 (and we recommend that the reader return to them in that context). In a nutshell, they allow us to show that if our testing algorithm accepts the environment $\texttt{ENV}$ with high constant probability, then there exists an environment $\texttt{ENV}'$ that evolves according to ρ and is relatively close to $\texttt{ENV}$. In particular, they aid us in defining the initial configuration $\texttt{ENV}'_0$ based on $\texttt{ENV}_t$ for some appropriate time step t.

▶ **Condition 5.** *Let $\sigma : \mathbb{Z}_n \to \{0,1\}$ be a configuration and let $[x,y]$ be an interval of locations such that $\sigma(\Gamma_k(x)) \in \mathcal{F}_\rho$ and $\sigma(\Gamma_k(y)) \in \mathcal{F}_\rho$. There exists a configuration $\widetilde{\sigma} \in \mathbb{Z}_n$, which differs from σ only on locations inside $[x,y]$, for which the following holds: For every $i \in [x,y]$ we have that $\widetilde{\sigma}(\Gamma_k(i)) \in \mathcal{F}_\rho$, and if $\sigma(\Gamma_k(i)) \in \mathcal{F}_\rho$, then $\widetilde{\sigma}(i) = \sigma(i)$.*

This condition also covers the special case in which $y = x$ and we interpret $[x,y]$ as $x, x+1, \ldots, x+n$ (with a slight abuse of notation).

▶ **Condition 6.** *Let $\sigma : \mathbb{Z}_n \to \{0,1\}$ be a configuration and $z \in \mathbb{Z}_n$ such that $\sigma(\Gamma_k(z)) \in \overline{\mathcal{F}}_\rho$. Let $\nu \in \{\tau_{k+1} : \tau \in \mathcal{F}_\rho\}$ and $\gamma, \gamma' \in \{0,1\}$. There exists a configuration $\widetilde{\sigma} : \mathbb{Z}_n \to \{0,1\}$ for which the following hold. There is a location $z' \in [z+1, z+2k+1]$ where $\widetilde{\sigma}(\Gamma_k(z')) \in \mathcal{F}_\rho$, and $f_\rho(\widetilde{\sigma}(z'), \gamma, \mathsf{parity}(z'-z) \oplus \gamma') = \nu$. Furthermore, for every $i \in [z+1, z'-1]$ it holds that $\widetilde{\sigma}(\Gamma_k(i)) \in \overline{\mathcal{F}}_\rho$, and for every $i \notin [z+k, z'+k]$, $\widetilde{\sigma}(i) = \sigma(i)$.*

A (symmetric) variant of the above should also hold if we replace $z' \in [z+1, z+2k+1]$ by $z' \in [z-2k-1, z-1]$, $i \in [z+1, z'-1]$ by $i \in [z'+1, z-1]$, and $i \notin [z+k, z'+k]$ by $i \notin [z'-k, z-k]$.

4 The Meta-Algorithm

In this section, we present a meta-algorithm for testing evolution of local rules that satisfy the sufficient conditions (specified in Section 3). Here we give an algorithm whose complexity is $\lceil n/m \rceil \cdot \mathrm{poly}(1/\epsilon)$ and, in the full version of this paper [9], we explain how to remove the dependence on n/m.

In order to precisely describe our meta-algorithm, we need to first define a particular set of locations that we designate as *the 1-dimensional grid* and the notion of *violating time-location pairs* with respect to the 1-dimensional grid. The 1-dimensional grid is defined in Section 4.1 and the notion of violating pairs is defined in Section 4.2. Then, in Section 4.3, we describe our meta-algorithm.

In the full version of this paper [9], we show that these conditions hold for all (non-trivial) threshold rules, as well as a couple of additional rules.

4.1 The grid

In this subsection we introduce the notion of a one-dimensional "grid", which will be a central building block of the meta algorithm (and its analysis). Recall that a configuration is a function $\sigma : \mathbb{Z}_n \to \{0,1\}$. A *partial* configuration is a function $\sigma' : \mathbb{Z}_n \to \{0,1\} \cup \perp$, which will serve us to denote restrictions of configurations to a subset of the locations.

Let $\Delta = \frac{\epsilon^2}{b_0} \cdot \min\{n, m\}$ where b_0 is a sufficiently large constant. We assume for simplicity that Δ and n/Δ are both integers. Let $G \subseteq \mathbb{Z}_n$ (the *grid*) be the set of locations $\{j \cdot (n/\Delta)\}_{j=0}^{n/\Delta - 1}$.

As we shall see in Section 4.3, our algorithm queries the tested environment on all grid locations and their k-neighborhoods at a specific time step t_1 (which will be set subsequently).

Let $\texttt{ENV}_t[G]$ be the partial configuration that agrees with $\texttt{ENV}_t$ on all locations in $\{\Gamma_k(g) : g \in G\}$ and is $\perp$ elsewhere.

▶ **Definition 7.** *Given a time step $t > 0$, we say that the partial configuration $\texttt{ENV}_t[G]$ induced by the k-neighborhoods of the grid locations at time t is feasible with respect to a rule ρ, if there exists an environment $\texttt{ENV}'$ that evolves according to ρ and such that $\texttt{ENV}'_t(i) = \texttt{ENV}_t(i)$ for every $i \in \Gamma_k(G)$. We say in such a case that $\texttt{ENV}'$ is a feasible completion of $\texttt{ENV}_t[G]$ with respect to ρ.*

▶ **Definition 8.** *Given a pair of grid locations $g_1, g_2 \in G$, a time step t and a subset $\mathcal{S} \subset \{0,1\}^{2k+1}$, if for every grid location $g \in G \cap [g_1, g_2]$ it holds that $\texttt{ENV}_t(\Gamma_k(g)) \in S$, then we say that the interval $[g_1, g_2]$ is an S grid interval with respect to $\texttt{ENV}_t$. We say that $[g_1, g_2]$ is a maximal $\mathcal{S}$ grid interval with respect to $\texttt{ENV}_t$, if both $\texttt{ENV}_t(g_1 - \Delta)$ and $\texttt{ENV}_t(g_2 + \Delta)$ do not belong to $\mathcal{S}$.*

In particular, we shall be interested in the case that $\mathcal{S}$ is $\mathcal{F}_\rho$ or $\overline{\mathcal{F}}_\rho$. Note that a grid interval $[g_1, g_2]$ contains all the locations between g_1 and g_2, and not just the grid locations. Also note that if $\texttt{ENV}_t(\Gamma_k(g)) \in S$ for every $g \in G$, then by Definition 8, there is no maximal $\mathcal{S}$ grid interval with respect to $\texttt{ENV}_t$ (we shall deal with such cases separately).

4.2 Violating Pairs

Let ρ be a fixed local rule that satisfies all the conditions stated in Section 3. Let $t_1 = \frac{b_1 \Delta}{\epsilon}$, where b_1 is a sufficiently large constant and Δ is as defined in Section 4.1. Let $t_2 = t_1 + \Delta$. We now define the concept of a violating pair $(t, i) \in [m] \times \mathbb{Z}_n$ with respect to $\texttt{ENV}_{t_1}$. Generally speaking, these are pairs in the environment $\texttt{ENV}$ whose values are inconsistent with evolving according to the rule ρ given the values at the grid locations at time t_1. The definition of a violating pair serves us later by allowing our algorithm to reject when it encounters one, which, as we prove, happens with high constant probability if $\texttt{ENV}$ is ϵ-far from evolving according to the rule ρ.

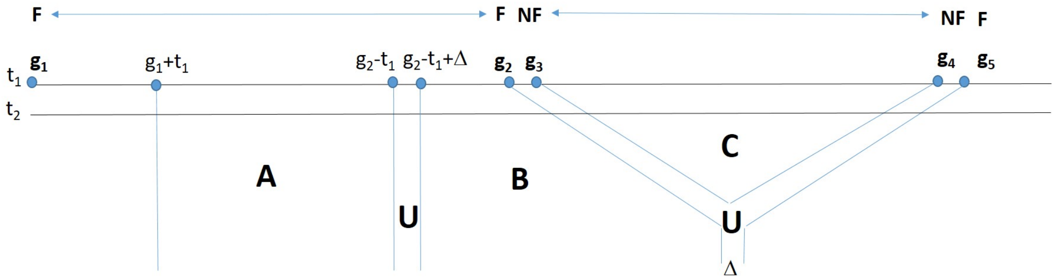

Figure 4.1 An illustration for the sets A, B, C, and U. Here $[g_1, g_2]$ is a maximal $\mathcal{F}_\rho$ grid interval, $g_3 = g_2 + \Delta$, where $[g_3, g_4]$ is a maximal $\overline{\mathcal{F}}_\rho$ grid interval, and $g_5 = g_4 + \Delta$ is an endpoint of a maximal $\mathcal{F}_\rho$ grid-interval. The area marked by A corresponds to pairs (t, i) such that $t > t_2$ and $i \in [g_1 + t_1, g_2 - t_1]$. These pairs are supposed to be final. The area marked by B corresponds to pairs (t, i) such that $t > t_2$, $i \in [g_2 - t_1 + \Delta, g_2 + (t - t_1)]$, and $\mathrm{dist}(g_2, i) < \mathrm{dist}(g_5, i) - \Delta$. These pairs are supposed to be final too. The area marked by C corresponds to pairs (t, i) such that $t > t_2$, $i \in [g_3, g_4]$, and (t, i) neither descends from $(t_1, g_3 + 1)$ nor from $(t_1, g_4 - 1)$. These pairs are supposed to be non-final. Finally, the areas marked by U correspond to pairs (t, i) such that $t > t_2$ and one of the following holds: **(1)** $i \in [g_2 - t_1 + 1, g_2(i) - t_1 + \Delta]$; **(2)** $i \in [g_3, g_4]$ and either **(a)** (t, i) descend from $(t_1, g_3 - \Delta)$ or $(t_1, g_4 + \Delta)$ and $|\mathrm{dist}(g_3, i) - \mathrm{dist}(g_4, i)| \le \Delta$, or **(b)** (t, i) does not descend from either $(t_1, g_3 - \Delta)$ or $(t_1, g_4 + \Delta)$ but it descends from either (t_1, g_3) or (t_1, g_4).

We next define three disjoint sets of time-location pairs, denoted A, B and C, and for each of these three sets we state conditions under which the pair is considered to be a violating pair with respect to $\texttt{ENV}_{t_1}[G]$. The proof that the three sets are pairwise disjoint appears in Section 5, and for an illustration, see Figure 4.1.

If $\texttt{ENV}_{t_1}(\Gamma_k(g)) \in \mathcal{F}_\rho$ for every $g \in G$, then $A = \{(t,i) : t_2 < t < m,\ i \in \mathbb{Z}_n\}$. Otherwise, A is the set of pairs (t,i) where $t_2 < t < m$ and $i \in \mathbb{Z}_n$ such that there exists a maximal $\mathcal{F}_\rho$ grid interval $[g_1(i), g_2(i)]$ with respect to $\texttt{ENV}_{t_1}$ for which $i \in [g_1(i)+t_1, g_2(i)-t_1]$.

▶ **Definition 9.** *A pair $(t,i) \in A$ is said to be a violating pair with respect to $\texttt{ENV}_{t_1}[G]$, if at least one of the following requirements does not hold. (1) $\texttt{ENV}_{t_2}(\Gamma_k(i)) \in \mathcal{F}_\rho$. (2) $\texttt{ENV}_t(\Gamma_k(i)) \in \mathcal{F}_\rho$. (3) $\texttt{ENV}_t(i) = f_\rho(\texttt{ENV}_{t_2}(i), \mathit{parity}(t-t_2), 0)$ where f_ρ is the function referred to in Condition 3.*

Let B be the set of pairs (t,i), where $t_2 < t < m$ and $i \in \mathbb{Z}_n$ for which the following holds. First, there exists a maximal $\mathcal{F}_\rho$ grid interval $[g_1(i), g_2(i)]$ with respect to $\texttt{ENV}_{t_1}$ such that either $i \in [g_1(i)-(t-t_1), g_1(i)+t_1-\Delta-1]$ or $i \in [g_2(i)-t_1+\Delta+1, g_2(i)+(t-t_1)]$. Second, for every other maximal $\mathcal{F}_\rho$ grid interval $[g'_1, g'_2]$ (with respect to $\texttt{ENV}_{t_1}$), if $i \in [g_1(i)-(t-t_1), g_1(i)+t_1-\Delta-1]$, then $\mathrm{dist}(g_1(i), i) < \mathrm{dist}(g'_2, i) - \Delta$, and if $i \in [g_2(i)-t_1+\Delta+1, g_2(i)+(t-t_1)]$, then $\mathrm{dist}(g_2(i), i) < \mathrm{dist}(g'_1, i) - \Delta$.

▶ **Definition 10.** *A pair $(t,i) \in B$ is said to be a violating pair with respect to $\texttt{ENV}_{t_1}[G]$, if at least one of the following requirements does not hold. (1) $\texttt{ENV}_t(\Gamma_k(i)) \in \mathcal{F}_\rho$. (2) Let $[g_1(i), g_2(i)]$ be the maximal $\mathcal{F}_\rho$ grid interval ensured by the definition of B given (t,i). Let $g(i)$ be the grid location in $G \cap ([g_1(i), g_1(i)+t_1-\Delta] \cup [g_2(i), g_2(i)-t_1+\Delta])$ that is closest to i (if there are two such grid locations, then select the one closer to $g_1(i)$). Then $\texttt{ENV}_t(i) = f_\rho(\texttt{ENV}_{t_1}(g(i)), \mathit{parity}(t-t_1), \mathit{parity}(\mathrm{dist}(i, g(i))))$, where f_ρ is the function referred to in Condition 3.*

If $\texttt{ENV}_{t_1}(\Gamma_k(g)) \in \overline{\mathcal{F}}_\rho$ for every $g \in G$, then $C = \{(t,i) : t_2 < t < m,\ i \in \mathbb{Z}_n\}$. Otherwise, C is the set of pairs (t,i) where $t_2 < t < m$ and $i \in \mathbb{Z}_n$ for which the following holds. First, there exists a maximal $\overline{\mathcal{F}}_\rho$ grid interval $[g_1(i), g_2(i)]$ with respect to $\texttt{ENV}_{t_1}$ such that $i \in [g_1(i), g_2(i)]$. Second, the pair (t,i) neither descends from the pair $(t_1, g_1(i)+1)$ nor from the pair $(t_1, g_2(i)-1)$.

▶ **Definition 11.** *A pair $(t,i) \in C$ is said to be a violating pair with respect to $\texttt{ENV}_{t_1}$, if at least one of the following requirements does not hold. (1) $\texttt{ENV}_t(\Gamma_k(i)) \in \overline{\mathcal{F}}_\rho$. (2) Let $g(i) \in G$ be a grid location satisfying $\mathrm{dist}(g(i), i) < \Delta$ (if there are two such grid locations, then select the one closer to $g_1(i)$). Then $\texttt{ENV}_t(\Gamma_k(i)) = h_\rho(\texttt{ENV}_{t_1}(\Gamma_k(g(i))), \mathit{parity}(t-t_1), \overrightarrow{\mathrm{dist}}(g(i), i))$.*

Finally, we define the set U of *uncertain* pairs (t,i), for which we cannot determine, given $\texttt{ENV}_{t_1}[G]$ and the corresponding pairs (t_2, i), whether they are violating or not.

▶ **Definition 12.** *The set U consists of all pairs $(t,i) \in \mathbb{Z}_n \times [m]$ such that $t > t_2$ and $(t,i) \notin A \cup B \cup C$.*

In Section 5 show that the number of pairs (t,i) belonging to the set U is relatively small, provided that $\texttt{ENV}_{t_1}[G]$ is feasible.

4.3 The testing algorithm

Recall that Let $\Delta = \frac{\epsilon^2}{b_0} \cdot \min\{n,m\}$, $t_1 = \frac{b_1 \Delta}{\epsilon}$, and $t_2 = t_1 + \Delta$ (where b_0 and b_1 are constants that will be set in the analysis).

Algorithm 1 The testing algorithm.

```
Tester for evolution according to a rule $\rho$
1. Query $\texttt{ENV}_{t_1}$ on all locations in $\Gamma_k(G)$. If $\texttt{ENV}_{t_1}[G]$ is infeasible with respect to $\rho$, reject.
2. Select uniformly at random $\Theta(\frac{1}{\epsilon})$ pairs $(t, i)$ where $i \in \mathbb{Z}_n$ and $t_2 < t < m$.
   For each selected pair $(t, i)$, query $\texttt{ENV}_t(\Gamma_k(i))$ and $\texttt{ENV}_{t_2}(\Gamma_k(i))$.
3. If some pair selected in Step 2 is a violating pair with respect to $\rho$, then reject.
   Otherwise, accept.
```

▶ **Theorem 3.** *Let ρ be any local rule that satisfies Conditions 1–6. Algorithm 1 is a one-sided error non-adaptive testing algorithm for evolution according to ρ whose query complexity is $O(\lceil n/m \rceil/\epsilon^2)$.*

The bound on the query complexity of the algorithm follows from the fact that the number of queries performed in Step 1 is $O(n/\Delta) = O(\lceil n/m \rceil/\epsilon^2)$ (recall that k is a constant), and the number of queries performed in Step 3 is $O(1/\epsilon)$. The correctness of the algorithm follows from the next two lemmas. We prove Lemma 1 in Section 6 and Lemma 2 in Section 7.

▶ **Lemma 1** (Completeness of the meta-algorithm). *Let ρ be any local rule that satisfies Conditions 1–6. If the environment* ENV *evolves according to ρ, then the algorithm accepts with probability 1.*

▶ **Lemma 2** (Soundness of the meta-algorithm). *Let ρ be any local rule that satisfies Conditions 1–6. If the environment* ENV *is ϵ-far from evolving according to ρ, then the algorithm rejects with probability at least $2/3$.*

5 Observations and simple claims

In this subsection we present several observations and simple claims that will be used in our proofs of Lemma 1 and Lemma 2.

The first two observations are directly implied by Conditions 1 and 2.

▶ **Observation 1.** *Let ρ be a local rule that satisfies Conditions 1 and 2, $\texttt{ENV} \in \mathcal{E}^{\rho}_{m,n}$ an environment that evolves according to ρ and $(t, i) \in [m] \times \mathbb{Z}_n$. If (t, i) has an ancestor (t', i') such that $\texttt{ENV}_{t'}(\Gamma_k(i')) \in \mathcal{F}_\rho$, then $\texttt{ENV}_t(\Gamma_k(i)) \in \mathcal{F}_\rho$.*

Note that Observation 1 implies that if $\texttt{ENV}_t(\Gamma_k(i)) \in \overline{\mathcal{F}}_\rho$, then $\texttt{ENV}_{t'}(\Gamma_k(i')) \in \overline{\mathcal{F}}_\rho$ for every ancestor (t', i') of (t, i).

▶ **Observation 2.** *Let ρ be a local rule that satisfies Conditions 1 and 2, $\texttt{ENV} \in \mathcal{E}^{\rho}_{m,n}$ an environment that evolves according to ρ and $(t, i) \in [m] \times \mathbb{Z}_n$, $t \leq n/2$. If $\texttt{ENV}_t(\Gamma_k(i)) \in \mathcal{F}_\rho$, then the location i belongs to an interval whose size is at least $2t$ such that $\texttt{ENV}'_t(\Gamma_k(j)) \in \mathcal{F}_\rho$ for every location j in this interval.*

The observation below directly follows from Observation 2 (as well as the definition of the grid G and Definitions 7 and 8).

▶ **Observation 3.** *Let ρ be a local rule that satisfies Conditions 1 and 2. Suppose that $\texttt{ENV}_t[G]$ for $t \leq n/2$ is feasible with respect to ρ. Then for every $[g_1, g_2]$ that is a maximal $\mathcal{F}_\rho$ grid interval with respect to $\texttt{ENV}_t$, the number of locations in $[g_1, g_2]$ is at least $2t - \Delta$.*

The next observation follows directly from Observation 1 (as well as the definition of G and Definitions 7 and 8).

▶ **Observation 4.** *Let ρ be a local rule that satisfies Conditions 1 and 2. Suppose that $\mathtt{ENV}_t[G]$ for $t \leq n/2$ is feasible with respect to ρ and let $g \in G$ be such that $\mathtt{ENV}_t(\Gamma_k(g)) \in \mathcal{F}_\rho$. If $\mathtt{ENV}'$ is a feasible completion of $\mathtt{ENV}_t[G]$ (with respect to ρ), then $\mathtt{ENV}'_{t'}(\Gamma_k(i)) \in \mathcal{F}_\rho$ for every $t' \geq t + \Delta$ and $i \in [g - \Delta, g + \Delta]$.*

Claim 5, stated next, also deals with feasible completions.

▷ Claim 5. Let ρ be a local rule that satisfies Conditions 1 and 2. Suppose that $\mathtt{ENV}_t[G]$ is feasible with respect to ρ for $t \geq \Delta$ and let $g \in G$ be such that both $\mathtt{ENV}_t(\Gamma_k(g)) \in \overline{\mathcal{F}}_\rho$ and $\mathtt{ENV}_t(\Gamma_k(g + \Delta)) \in \overline{\mathcal{F}}_\rho$. If $\mathtt{ENV}'$ is a feasible completion of $\mathtt{ENV}_t[G]$ (with respect to ρ), then $\mathtt{ENV}'_t(\Gamma_k(i)) \in \overline{\mathcal{F}}_\rho$ for every $i \in [g, g + \Delta]$.

Recall the definitions of the sets A, B and C from Section 4.2.

▷ Claim 6. The sets A, B, and C are pairwise disjoint.

In the last claim of this subsection, we bound the size of the set U of uncertain pairs (as defined in Definition 12).

▷ Claim 7. If $\mathtt{ENV}_{t_1}[G]$ is feasible (with respect to ρ), then $|U| \leq \frac{5\epsilon}{b_1} mn$ (where b_1 is the constant in the setting of $t_1 = \frac{b_1 \Delta}{\epsilon}$).

We note that Claim 7 does not depend on the setting of Δ, but only on the definition of t_1 as a function of Δ (as well as the definition of the grid G, which, too is defined based on Δ, and in turn is used to determine U).

6 Proof of Lemma 1: Completeness of the meta-algorithm

Let ρ be any local rule that satisfies Conditions 1–6 (where in this proof we do not make use of Conditions 5 and 6, which are provided in the next subsection), and let $\mathtt{ENV} \in \mathcal{E}^{\rho}_{m,n}$ be a dynamic environment that evolves according to ρ. The only steps in which our algorithm may reject are Step 1 and Step 3. The grid is feasible by definition, and hence the algorithm does not reject in Step 1. To show that it also does not reject in Step 3, we show that there are no violating pairs with respect to $\mathtt{ENV}_{t_1}[G]$. Recall that each violating pair belongs to one of the three sets A, B, or C (as defined in Section 4.2). Specifically, we next show that in each of the three cases ($(t, i) \in A$, $(t, i) \in B$, and $(t, i) \in C$), the requirements (specified in Section 4.2) for (t, i) being a non-violating pair hold. In what follows, if we say that a pair (t, i) is final (similarly, non-final), then we mean with respect to $\mathtt{ENV}$, and when we refer to maximal grid intervals, it is always with respect to $\mathtt{ENV}_{t_1}$, and violations are always with respect to $\mathtt{ENV}_{t_1}[G]$.

Pairs $(t, i) \in A$. By the definition of A, $t > t_2$ and there exists a grid location $g(i) \in G$ such that $\mathrm{dist}(i, g(i)) \leq \Delta$ and $\mathtt{ENV}_{t_1}(\Gamma_k(g(i))) \in \mathcal{F}_\rho$ (this holds both in the case that $\mathtt{ENV}_{t_1}(\Gamma_k(g)) \in \mathcal{F}_\rho$ for every $g \in G$ and in the case that there exists a maximal $\mathcal{F}_\rho$ grid interval $[g_1(i), g_2(i)]$ such that $i \in [g_1(i) + t_1, g_2(i) - t_1]$.) By Observation 4, both $\mathtt{ENV}_{t_2}(\Gamma_k(i)) \in \mathcal{F}_\rho$ and $\mathtt{ENV}_t(\Gamma_k(i)) \in \mathcal{F}_\rho$. Turning to the third requirement, by Condition 3, applied with $t' = t_2$ and $i' = i$, we get that $\mathtt{ENV}_t(i) = f_\rho(\mathtt{ENV}_{t_2}(i), \mathsf{parity}(t - t_2), 0)$. Therefore, all three requirements on pairs in A hold, and hence (t, i) is not a violating pair.

Pairs $(t, i) \in B$. By the definition of B, $t > t_2$ and there exists a maximal $\mathcal{F}_\rho$ grid interval $[g_1(i), g_2(i)]$ with respect to $\mathtt{ENV}_{t_1}$ such that either $i \in [g_1(i) - (t - t_1), g_1(i) + t_1 - \Delta - 1]$ or $i \in [g_2(i) - t_1 + \Delta + 1, g_2(i) + (t - t_1)]$. Furthermore, for every other maximal $\mathcal{F}_\rho$ grid interval $[g'_1, g'_2]$ (with respect to $\mathtt{ENV}_{t_1}$), if $i \in [g_1(i) - (t - t_1), g_1(i) + t_1 - \Delta - 1]$, then $\mathrm{dist}(g_1(i), i) < \mathrm{dist}(g'_2, i) - \Delta$, and if $i \in [g_2(i) - t_1 + \Delta + 1, g_2(i) + (t - t_1)]$, then $\mathrm{dist}(g_2(i), i) < \mathrm{dist}(g'_1, i) - \Delta$. Let $g(i)$ be the grid location closest to i in $G \cap ([g_1(i), g_1(i) + t_1 - \Delta] \cup [g_2(i) - t_1 + \Delta, g_2(i)])$ (as defined in the second requirement concerning (non-)violating pairs $(t, i) \in B$).

We claim that (t, i) descends from $(0, g(i))$. To see why, first consider the case in which $i \in [g_1(i) - (t - t_1), g_1(i)] \cup [g_2(i), g_2(i) + (t - t_1)]$. In this case, either $g(i) = g_1(i)$ or $g(i) = g_2(i)$, which means that $\mathrm{dist}(i, g(i)) \leq t - t_1 \leq t$. Second, consider the case in which $i \in [g_1(i), g_1(i) + t_1 - \Delta - 1] \cup [g_2(i) - t_1 + \Delta + 1, g_2(i)]$. In this case, the grid location closest to i in $G \cap ([g_1(i), g_1(i) + t_1 - \Delta] \cup [g_2(i) - t_1 + \Delta, g_2(i)])$ is within a distance of at most Δ from the location i. Hence, $\mathrm{dist}(i, g(i)) \leq \Delta \leq t$. Therefore, in any case, $\mathrm{dist}(i, g(i)) \leq t$, and thus the pair (t, i) descends from the pair $(0, g(i))$.

Assume (without loss of generality) that $g(i) \in [g_2(i) - t_1 + \Delta, g_2(i)]$. Since $\mathtt{ENV}_{t_1}(\Gamma_k(g_2(i) + \Delta)) \in \overline{\mathcal{F}}_\rho$ (as $[g_1(i), g_2(i)]$ is a maximal final grid interval), we know (by Observation 1) that $\mathtt{ENV}_0(\Gamma_k(j)) \in \overline{\mathcal{F}}_\rho$ for every $j \in [g_2(i) + \Delta - t_1, g_2(i) + \Delta + t_1]$. However, since $\mathtt{ENV}_{t_1}(\Gamma_k(g_2(i)) \in \mathcal{F}_\rho$, there must be some location $\ell \in [g_2(i) - t_1, g_2(i) + \Delta - t_1 - 1]$ such that $\mathtt{ENV}_0(\Gamma_k(\ell)) \in \mathcal{F}_\rho$. Among the locations ℓ that satisfy these conditions, let ℓ^* be the one that minimizes $\mathrm{dist}(\ell, g_2(i) + \Delta - t_1)$, so that for every $\ell' \in [\ell^* + 1, g_2(i) + \Delta - t_1]$ we have that $\mathtt{ENV}_0(\Gamma_k(\ell')) \in \overline{\mathcal{F}}_\rho$. Hence, for every $i'' \neq g(i)$ satisfying $\mathrm{dist}(g(i), i'') \leq \mathrm{dist}(g(i), \ell^*)$ it holds that $\mathtt{ENV}_0(i'') \in \overline{\mathcal{F}}_\rho$. Additionally, since $g(i) \in [g_2(i) - t_1 + \Delta, g_2(i)]$ and $\ell^* \in [g_2(i) - t_1, g_2(i) + \Delta - t_1 - 1]$, it must hold that $\mathrm{dist}(g(i), \ell^*) \leq t_1$, which means that the pair $(t_1, g(i))$ descends from the pair $(0, \ell^*)$. Also, both $\mathtt{ENV}_{t_1}(g(i)) \in \mathcal{F}_\rho$ and $\mathtt{ENV}_0(\ell^*) \in \mathcal{F}_\rho$. Thus, we can apply Condition 3 for the two pairs $(0, \ell^*)$ and $(t_1, g(i))$ to get that $\mathtt{ENV}_{t_1}(g(i)) = f_\rho(\mathtt{ENV}_0(\ell^*), \mathsf{parity}(t_1), \mathsf{parity}(\mathrm{dist}(\ell^*, g(i))))$.

Since the pair (t, i) descends from the pair $(t_1, g(i))$, and the pair $(t_1, g(i))$ descends from the pair $(0, \ell^*)$, it holds that the pair (t, i) must also descend from the pair $(0, \ell^*)$. Additionally, both both $\mathtt{ENV}_t(i) \in \mathcal{F}_\rho$ and $\mathtt{ENV}_0(\ell^*) \in \mathcal{F}_\rho$. Also, by the second requirement on (t, i), involving other maximal $\mathcal{F}_\rho$ grid intervals $[g'_1, g'_2]$, for every $i'' \neq i$ satisfying $\mathrm{dist}(i, i'') \leq \mathrm{dist}(i, \ell^*)$ it holds that $\mathtt{ENV}_0(i'') \in \overline{\mathcal{F}}_\rho$. Thus, we can apply Condition 3 for the two pairs $(0, \ell^*)$ and (t, i) to get that $\mathtt{ENV}_t(i) = f_\rho(\mathtt{ENV}_0(\ell^*), \mathsf{parity}(t), \mathsf{parity}(\mathrm{dist}(\ell^*, i)))$. But then, since f_ρ is the XOR of its first argument and a subset of the other two, and $\mathsf{parity}(t - t_1) = \mathsf{parity}(t_1) \oplus \mathsf{parity}(t)$ as well as $\mathsf{parity}(\mathrm{dist}(g(i), i))) = \mathsf{parity}(\mathrm{dist}(\ell^*, g(i))) \oplus \mathsf{parity}(\mathrm{dist}(\ell^*, i))$, we get that $\mathtt{ENV}_t(i) = f_\rho(\mathtt{ENV}_{t_1}(g(i)), \mathsf{parity}(t - t_1), \mathsf{parity}(\mathrm{dist}(g(i), i)))$.

Pairs $(t, i) \in C$. There are two cases (where in both $t > t_2$). The first is that $\mathtt{ENV}_{t_1}(\Gamma_k(g)) \in \overline{\mathcal{F}}_\rho$ for every $g \in G$ (so that i may be any location in $\mathbb{Z}_n$). In the second case there exists a maximal $\overline{\mathcal{F}}_\rho$ grid interval $[g_1(i), g_2(i)]$ such that $i \in [g_1(i), g_2(i)]$, and (t, i) does not descend from either $(t_1, g_1(i) - 1)$ or $(t_1, g_2(i) + 1)$, which implies that for every $j \in \mathbb{Z}_n$, if the pair (t, i) descends from (t_1, j), then $j \in [g_1(i), g_2(i)]$. In both cases, by Claim 5, all ancestors (t_1, j) of (t, i) satisfy $\mathtt{ENV}_{t_1}(\Gamma_k(j)) \in \overline{\mathcal{F}}_\rho$. By Observation 1 this implies that $\mathtt{ENV}_t(\Gamma_k(i)) \in \overline{\mathcal{F}}_\rho$, so that the first requirement is met. As for the second requirement, since the grid location $g(i)$ defined in the second requirement is such that $(t_1, g(i))$ is an ancestor of (t, i) (and $\mathtt{ENV}_{t_1}(\Gamma_k(g(i))) \in \overline{\mathcal{F}}_\rho$), we can apply Condition 4 (with $t' = t_1$ and $i' = g$) to get that $\mathtt{ENV}_t(\Gamma_k(i)) = h_\rho(\mathtt{ENV}_{t_1}(\Gamma_k(g(i))), \mathsf{parity}(t - t_1), \overrightarrow{\mathrm{dist}}(g(i), i)$, as required.

We've shown that under the premise of the lemma, there is no pair $(t, i) \in A \cup B \cup C$ that is a violating pair. Thus, our algorithm cannot reject at Step 3.

7 Proof of Lemma 2: Soundness of the meta-algorithm

Let ENV be any environment that is ϵ-far from evolving according to ρ, where ρ is a local rule that satisfies Conditions 1–6. If $\mathtt{ENV}_{t_1}[G]$ is infeasible with respect to ρ, then the algorithm rejects (in Step 1). Hence, we assume from now on that $\mathtt{ENV}_{t_1}[G]$ is feasible.

We claim that the number of violating pairs with respect to $\mathtt{ENV}_{t_1}[G]$ is at least $\frac{\epsilon}{b_2}mn$, where $b_2 > 1$ is a constant. Lemma 2 follows, since the algorithm selects $s = \Theta(1/\epsilon)$ pairs (in Step 2), and rejects if any of them is found to be a violating pair (in Step 3). Hence, the probability that the algorithm rejects is at least $1 - (1 - \epsilon/b_2)^s$, which is at least $2/3$ for $s \geq 2b_2/\epsilon$.

Suppose by way of contradiction that there are less than $\frac{\epsilon}{b_2}mn$ violating pairs. We show how, based on ENV (to be precise, $\mathtt{ENV}_{t_1}[G]$ and $\mathtt{ENV}_{t_2}$) we can define an environment $\mathtt{ENV}'$ for which the following holds. First, $\mathtt{ENV}'$ evolves according to ρ. Second, $\mathtt{ENV}'$ differs from ENV on at most ϵmn pairs $(t, i) \in \mathbb{Z}_n \times [m]$. But this contradicts the premise that ENV is ϵ-far from evolving according to ρ. Details follow in the next subsections.

We first provide all details (in Section 7.1 and Section 7.2) under the assumption that there exist grid locations $g \in G$ for which $\mathtt{ENV}_{t_1}(\Gamma_k(g)) \in \mathcal{F}_\rho$ as well as grid locations $g' \in G$ for which $\mathtt{ENV}_{t_1}(\Gamma_k(g')) \in \overline{\mathcal{F}}_\rho$. We discuss (in the full version of this paper [9]) the two special cases for which either $\mathtt{ENV}_{t_1}(\Gamma_k(g)) \in \mathcal{F}_\rho$ for every $g \in G$ or $\mathtt{ENV}_{t_1}(\Gamma_k(g)) \in \overline{\mathcal{F}}_\rho$ for every $g \in G$, which we refer to as the *homogeneous* cases.

7.1 The definition of $\mathtt{ENV}'$

To construct the dynamic environment $\mathtt{ENV}'$, we define its initial configuration $\mathtt{ENV}'_0$, and then apply the local rule ρ for $m - 1$ steps. Hence, $\mathtt{ENV}'$ evolves according to ρ by construction. The initial configuration $\mathtt{ENV}'_0$ is defined with respect to a configuration σ on which we perform several transformations to obtain $\mathtt{ENV}'_0$. We define the configuration σ by specifying the value of $\sigma(i)$ for each location $i \in \mathbb{Z}_n$ as explained next. In what follows, whenever we refer to maximal $\overline{\mathcal{F}}_\rho$ grid intervals (similarly, maximal $\mathcal{F}_\rho$ grid intervals), it is with respect to $\mathtt{ENV}_{t_1}$.

We shall make use of a function $h_\rho^{\leftarrow} : \overline{\mathcal{F}}_\rho \times \{0, 1\} \times \mathbb{Z}_n$ (based on h_ρ – see Condition 4). Recall that by Condition 4, for each fixed $\tau \in \overline{\mathcal{F}}_\rho$, $\beta \in \{0, 1\}$ and $\ell \in \mathbb{Z}_n$, there exists a unique τ' such that $h_\rho(\tau', \beta, \ell) = \tau$.

▶ **Definition 8.** *For any $\tau \in \overline{\mathcal{F}}_\rho$, $\beta \in \{0, 1\}$ and $\ell \in \mathbb{Z}_n$, $h_\rho^{\leftarrow}(\tau, \beta, \ell)$ equals the (unique) pattern τ' for which $h_\rho(\tau', \beta, \ell) = \tau$.*

We also make the following observation, based on Condition 3, by which f_ρ is the XOR of its first argument and a subset of the other two.

▶ **Observation 9.** *For any $\beta_1, \beta_2, \beta_3 \in \{0, 1\}$, if $f_\rho(\beta_1, \beta_2, \beta_3) = \beta'_1$, then $f_\rho(\beta'_1, \beta_2, \beta_3) = \beta_1$. Furthermore, for any $\beta'_2, \beta'_3 \in \{0, 1\}$, $f_\rho(f_\rho(\beta_1, \beta_2, \beta_3), \beta'_2, \beta'_3) = f_\rho(\beta_1, \beta_2 \oplus \beta'_2, \beta_3 \oplus \beta'_3)$, and in particular, $f_\rho(f_\rho(\beta_1, \beta_2, \beta_3), \beta_2, \beta_3) = \beta_1$.*

For each maximal $\overline{\mathcal{F}}_\rho$ grid interval $[g_1, g_2]$, let $J(g_1, g_2) = [g_1 - t_1 - k, g_2 + t_1 + k]$ and let J be the union over all such sets. We also define $J_1(g_1, g_2) = [g_1 - t_1, g_2 + t_1]$ (for each $\overline{\mathcal{F}}_\rho$ grid interval $[g_1, g_2]$), and let $J_1 \subset J$ be the union over all such sets.

We first establish two simple claims.

▷ Claim 10. Let ρ be any local rule that satisfies Conditions 1–4. For every two maximal $\overline{\mathcal{F}}_\rho$ grid interval $[g_1, g_2]$ and $[g'_1, g'_2]$, we have that $J(g_1, g_2) \cap J(g'_1, g'_2) = \emptyset$.

▷ Claim 11. Let ρ be any local rule that satisfies Conditions 1–4. Let $\texttt{ENV}''$ be any environment that is a feasible extension of $\texttt{ENV}_{t_1}[G]$ with respect to ρ, and let $[g_1, g_2]$ be a maximal $\overline{\mathcal{F}}_\rho$ grid interval (with respect to $\texttt{ENV}_{t_1}[G]$). Then $\texttt{ENV}''_0(\Gamma_k(i)) \in \overline{\mathcal{F}}_\rho$ for every $i \in J_1(g_1, g_2)$. Furthermore, $\texttt{ENV}''_0(\Gamma_k(i)) = h^{\leftarrow}_\rho(\texttt{ENV}_{t_1}(\Gamma_k(g)), \mathsf{parity}(t_1), \overrightarrow{\mathrm{dist}}(i, g))$ for any $g \in G \cap [g_1, g_2]$ and every ancestor $(0, i)$ of (t_1, g).

Observe that Claim 11 implies that $\texttt{ENV}''_0$ is *uniquely* determined by $\texttt{ENV}_{t_1}[G]$ on all location in J for every $\texttt{ENV}''$ that is a feasible extension of $\texttt{ENV}_{t_1}[G]$ (with respect to ρ). Based on this observation, we start by setting the locations of σ that belong to J as in such $\texttt{ENV}''_0$. In particular we have that $\sigma(\Gamma_k(i)) \in \overline{\mathcal{F}}_\rho$ for every $i \in J_1$, and furthermore,

$$\forall i \in J_1, g \in G \text{ s.t. } (t_1, g) \text{ descends from } (0, i) \text{ and,}$$
$$\sigma(\Gamma_k(i)) = h^{\leftarrow}_\rho(\texttt{ENV}_{t_1}(\Gamma_k(g)), \mathsf{parity}(t_1), \overrightarrow{\mathrm{dist}}(i, g)) \ . \tag{7.1}$$

Turning to the locations not yet set in σ, for each location $i \in \mathbb{Z}_n \setminus J$, $\sigma(i) = f_\rho(\texttt{ENV}_{t_2}(i), \mathsf{parity}(t_2), 0)$. Note that by Observation 9, $f_\rho(\sigma(i), \mathsf{parity}(t_2), 0) = \texttt{ENV}_{t_2}(i)$.

We next explain how we modify σ so as to obtain $\texttt{ENV}'_0$ using Condition 5 and Condition 6. The modifications are performed (strictly) within the following set of intervals $\mathcal{S}$.

$$\mathcal{S} = \left\{ \left[a = g_1 - \Delta + t_1, b = g_2 + \Delta - t_1\right] \ : \ [g_1, g_2] \text{ is a maximal } \mathcal{F}_\rho \text{ grid interval } \right\} \ . \tag{7.2}$$

The intervals in $\mathcal{S}$ are clearly disjoint (as each is a sub-interval of a different maximal $\mathcal{F}_\rho$ grid interval), and by Observation 3, each is non-empty. Note that for each maximal $\mathcal{F}_\rho$ grid interval $[g_1, g_2]$, we have that $g_1 - \Delta$ and $g_2 + \Delta$ are endpoints of maximal $\overline{\mathcal{F}}_\rho$ grid interval. Therefore, $a, b \in J_1$ for each interval $[a, b] \in \mathcal{S}$, and by the setting of σ and Claim 11, $\sigma(\Gamma_k(a)), \sigma(\Gamma_k(b)) \in \overline{\mathcal{F}}_\rho$.

For each $[a, b] \in \mathcal{S}$ and the corresponding $[g_1, g_2]$, let $\alpha(a, b) = \texttt{ENV}_{t_1}(g_1)$, $\beta(a, b) = \texttt{ENV}_{t_1}(g_2)$, $\gamma(a, b) = \mathsf{parity}(t_1)$, $\gamma'(a, b) = \mathsf{parity}(t_1 - \Delta)$. We shall apply Condition 5 and Condition 6 to modify σ on all $[a, b] \in \mathcal{S}$ “in parallel” as described next, and set $\texttt{ENV}'_0$ to be the resulting configuration.

For each $[a, b] \in \mathcal{S}$ we first apply Condition 6 with $z = a$, $\nu = \alpha(a, b)$, $\gamma = \gamma(a, b)$ and $\gamma' = \gamma'(a, b)$. We let $a' = z'$ (recall that $z' \in [z+1, z+2k+1]$ and $\widetilde{\sigma}(\Gamma_k(z')) \in \mathcal{F}_\rho$). Next we apply Condition 6 in its second (symmetric) variant with $z = b$, $\nu = \beta(a, b)$, $\gamma = \gamma(a, b)$ and $\gamma' = \gamma'(a, b)$. We let $b' = z'$ (recall that in this variant, $z' \in [z - 2k - 2, z - 1]$, and here too $\widetilde{\sigma}(\Gamma_k(z')) \in \mathcal{F}_\rho$). Finally we apply Condition 5 on the modified configuration with $x = a'$ and $y = b'$.

7.2 The distance between ENV and ENV′

In this subsection we show that based on the counter-assumption regarding the number of violating pairs, the number of pairs $(t, i) \in \mathbb{Z}_n \times [m]$ on which $\texttt{ENV}$ and $\texttt{ENV}'$ differ is at most ϵmn. To this end we show that each (t, i) such that $\texttt{ENV}_t(i) \neq \texttt{ENV}'_t(i)$ belongs to one of the following sets:

1. The set of pairs (t, i) for which $0 \leq t \leq t_2$.
2. The uncertainty set U.
3. The set of (t, i) pairs where (t, i) is a violation with respect to $\texttt{ENV}_{t_1}[G]$.

By the setting of t_2 (t_1) and Δ, the number of pairs in the first set is at most $\frac{(b_1+1)\epsilon}{b_0} mn$. By Claim 7, $|U| \leq \frac{5\epsilon}{b_1} mn$. By our counter-assumption, the number of violating pairs is at most $\frac{\epsilon}{b_2} mn$. Setting $b_1 = 15$, $b_0 = 48$ and $b_2 = 3$, we get a total of at most ϵmn pairs, as claimed.

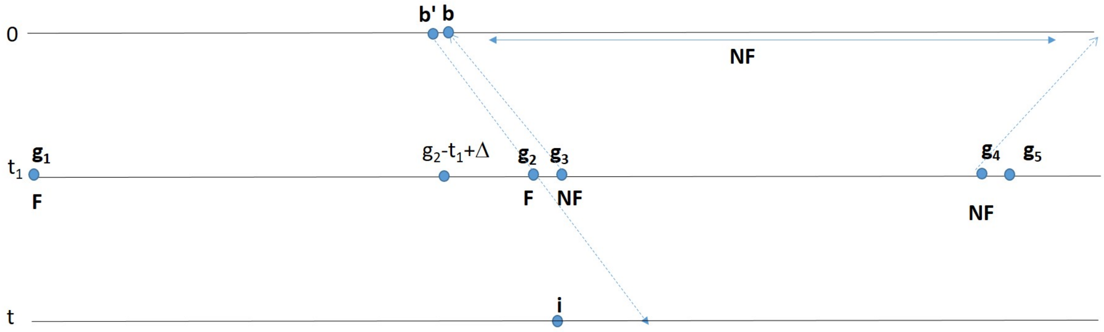

Figure 7.1 An illustration for the setting of $\texttt{ENV}_0$. As in Figure 4.1, $[g_1, g_2]$ is a maximal $\mathcal{F}_\rho$ grid interval, $g_3 = g_2 + \Delta$, where $[g_3, g_4]$ is a maximal $\overline{\mathcal{F}}_\rho$ grid interval, and $g_5 = g_4 + \Delta$ is an endpoint of a maximal $\mathcal{F}_\rho$ grid-interval. The maximal $\overline{\mathcal{F}}_\rho$ grid interval $[g_3, g_4]$ is used to set the locations between $g_3 - t_1 = g_2 - t_1 + \Delta$ and $g_4 + t_1$ (more precisely, between $g_3 - t_1 - k$ and $g_4 + t_1 + k$ based on Claim 11. The location $b = g_2 - t_1 + \Delta$ is an endpoint of an interval in $\mathcal{S}$, and the location b' is determined by the application of Condition 6. The values in the k-neighborhood of b' are set so that the evolution of ρ will result in the $\texttt{ENV}_{t_1}(g_2)$ at time t_1. The pair (t, i) belongs to the set B.

To establish the claim that each (t, i) for which $\texttt{ENV}_t(i) \neq \texttt{ENV}'_t(i)$ belongs to one of the above three sets, we prove the contrapositive. Suppose the pair (t, i) is not in the uncertainty set U and that $t > t_2$. It follows that $(t, i) \in A \cup B \cup C$. We show that for each of the three types of pairs ($(t, i) \in A$, $(t, i) \in B$, and $(t, i) \in C$), if the pair (t, i) is not a violating pair with respect to $\texttt{ENV}_{t_1}[G]$, it must hold that $\texttt{ENV}_t(i) = \texttt{ENV}'_t(i)$.

Pairs $(t, i) \in A$. By the definition of A, there exists a maximal $\mathcal{F}_\rho$ grid interval $[g_1(i), g_2(i)]$ (with respect to $\texttt{ENV}_{t_1}$) for which $i \in [g_1(i) + t_1, g_2(i) - t_1]$. Since (t, i) is not a violating pair with respect to $\texttt{ENV}_{t_1}[G]$, it must hold that $\texttt{ENV}_{t_2}(\Gamma_k(i)), \texttt{ENV}_t(\Gamma_k(i)) \in \mathcal{F}_\rho$ and that $\texttt{ENV}_t(i) = f_\rho(\texttt{ENV}_{t_2}(i), \mathsf{parity}(t - t_2), 0)$. Since $i \in [g_1(i) + t_1, g_2(i) - t_1]$, we know that $i \notin J$. Hence, by the definition of the configuration σ, we have that $\sigma(i) = f_\rho(\texttt{ENV}_{t_2}(i), \mathsf{parity}(t_2), 0)$ and that $\sigma(\Gamma_k(i)) \in \mathcal{F}_\rho$. Let $[a(i), b(i)] = [g_1(i) - \Delta + t_1, g_2(i) + \Delta - t_1]$, so that $i \in I(a(i), b(i))$. By the definition of E'_0, based on Condition 5 we have that $\texttt{ENV}'_0(i) = \sigma(i)$ and $\texttt{ENV}'_0(\Gamma_k(i)) \in \mathcal{F}_\rho$. Since $\texttt{ENV}'$ evolves according to ρ, by Condition 3,

$$\texttt{ENV}'_{t_2}(i) = f_\rho(\texttt{ENV}'_0(i), \mathsf{parity}(t_2), 0) = f_\rho(f_\rho(\texttt{ENV}_{t_2}(i), \mathsf{parity}(t_2), 0), \mathsf{parity}(t_2), 0) = \texttt{ENV}_{t_2}(i)$$

where the last equality follows from (the second part of) Observation 9. Additionally, by Condition 1, $\texttt{ENV}'_{t_2}(i) \in \mathcal{F}_\rho$. Therefore, by Condition 3,

$$\texttt{ENV}'_t(i) = f_\rho(\texttt{ENV}'_{t_2}(i), \mathsf{parity}(t - t_2), 0) = f_\rho(\texttt{ENV}_{t_2}(i), \mathsf{parity}(t - t_2), 0) = \texttt{ENV}_t(i) \ .$$

Pairs $(t, i) \in B$. By the definition of B, there exists a maximal $\mathcal{F}_\rho$ grid interval $[g_1(i), g_2(i)]$ with respect to $\texttt{ENV}_{t_1}$ such that either $i \in [g_1(i) - (t - t_1), g_1(i) + t_1 - \Delta - 1]$ or $i \in [g_2(i) - t_1 + \Delta + 1, g_2(i) + (t - t_1)]$. Assume (without loss of generality) that the latter holds. By the definition of B we also know that for every other maximal $\mathcal{F}_\rho$ grid interval $[g'_1, g'_2]$ it holds that $\text{dist}(i, g_2(i)) < \text{dist}(i, g'_1), \text{dist}(i, g'_2) - \Delta$. Let $g(i)$ be the grid location closest to i in $[g_2(i) - t_1 + \Delta, g_2(i)]$. Since $i \in [g_2(i) - t_1 + \Delta + 1, g_2(i) + (t - t_1)]$, necessarily, $g(i) \in [g_2(i) - t_1, g_2(i)]$. For the sake of conciseness, in what follows we shall use g_1, g_2, and g as a shorthand for $g_1(i)$, $g_2(i)$ and $g(i)$, respectively.

Since $[g_1, g_2]$ is a maximal $\mathcal{F}_\rho$ grid interval, $[a = g_1 - \Delta + t_1, b = g_2 + \Delta - t_1] \in \mathcal{S}$. Hence, to obtain $\mathtt{ENV}'_0$ from the configuration σ, we invoked Condition 6 (the symmetric version) with $z = b$, $\nu = \mathtt{ENV}_{t_1}(g_2)$, $\gamma = \mathsf{parity}(t_1)$, and $\gamma' = \mathsf{parity}(t_1 - \Delta) = \mathsf{parity}(\mathrm{dist}(g_2, b))$. By Condition 6, letting $b' = z'$, $\mathtt{ENV}_{t_1}(g_2) = f_\rho(\mathtt{ENV}'_0(b'), \mathsf{parity}(t_1), \mathsf{parity}(\mathrm{dist}(g_2, b')))$. By Observation 9,

$$\mathtt{ENV}'_0(b') = f_\rho^{\leftarrow}(\mathtt{ENV}_{t_1}(g_2), \mathsf{parity}(t_1), \mathsf{parity}(\mathrm{dist}(g_2, b'))) \ . \tag{7.3}$$

As $b' \in [b - 2k - 1, b - 1]$, which by the setting of b implies that $b' \in [g_2 + \Delta - t_1 - 2k - 1, g_2 + \Delta - t_1 - 1]$, we have that $(0, b')$ is an ancestor of (t_1, j) for every $j \in [g_2 - t_1, g_2]$. In particular this holds for the grid location g (that is closest to i in $G \cap [g_2 - t_1, g_2]$). Since (t, i) descends from (t_1, g), we get that (t, i) also descends from $(0, b')$.

We claim that for every $b'' \neq b'$ with $\mathrm{dist}(i, b'') < \mathrm{dist}(i, b')$ it holds that $\mathtt{ENV}'_0(b'') \in \overline{\mathcal{F}}_\rho$. To verify this, let $[g_3, g_4]$ be the maximal $\overline{\mathcal{F}}_\rho$ grid interval where $g_3 = g_2 + \Delta$, and let $g_5 = g_4 + \Delta$, so that g_5 is the endpoint of a maximal $\mathcal{F}_\rho$ grid interval. By the definition of $\mathtt{ENV}'_0$ (based on σ and Condition 6) we have that $\mathtt{ENV}'_0(\Gamma_k(j)) \in \overline{\mathcal{F}}_\rho$ for every $j \in [b' + 1, b - 1] \cup J_1(g_3, g_4) = [b' + 1, g_4 + t_1]$. Since (by the second requirements on pairs in B), $\mathrm{dist}(i, g_2) < \mathrm{dist}(i, g_5) - \Delta$ and $b' \in [g_2 - t_1 + \Delta - 2k - 1, g_2 - t_1 + \Delta - 1]$, we have that $\mathtt{ENV}'_0(b'') \in \overline{\mathcal{F}}_\rho$ for every $b'' \neq b'$ with $\mathrm{dist}(i, b'') < \mathrm{dist}(i, b')$. Therefore, we can apply Condition 3 to obtain that

$$\begin{aligned} \mathtt{ENV}'_t(i) &= f_\rho(\mathtt{ENV}_0(b'), \mathsf{parity}(t), \mathsf{parity}(\mathrm{dist}(i, b'))) \\ &= f_\rho(f_\rho(\mathtt{ENV}_{t_1}(g_2), \mathsf{parity}(t_1), \mathsf{parity}(\mathrm{dist}(g_2, b'))), \mathsf{parity}(t), \mathsf{parity}(\mathrm{dist}(i, b'))) && (7.4) \\ &= f_\rho(\mathtt{ENV}_{t_1}(g_2), \mathsf{parity}(t - t_1), \mathsf{parity}(\mathrm{dist}(i, g_2))) && (7.5) \end{aligned}$$

where the last equality follows from Observation 9.

Consider first the case that $g = g_2$. Since the pair (t, i) is not a violating pair,

$$\mathtt{ENV}_t(i) \ = \ f_\rho(\mathtt{ENV}_{t_1}(g_2), \mathsf{parity}(t - t_1), \mathsf{parity}(\mathrm{dist}(i, g_2))) \ , \tag{7.6}$$

and hence in this case, $\mathtt{ENV}_t(i) = \mathtt{ENV}'_t(i)$, as desired. We next turn to the case that $g \neq g_2$. We claim that since $\mathtt{ENV}_{t_1}[G]$ is feasible,

$$\mathtt{ENV}_{t_1}(g) = f_\rho(\mathtt{ENV}'_0(b'), \mathsf{parity}(t_1), \mathsf{parity}(\mathrm{dist}(g, b'))) \ . \tag{7.7}$$

Conditioned on Equality 7.7 holding, the argument is the same as for the case that $g = g_2$ (replacing g_2 with g in Equations (7.4)–(7.6)).

To verify Equation (7.7), we introduce the notion of a *source* for a final pair. Let $\mathtt{ENV}''$ be an environment that evolves according to ρ, and (t', i') a final pair with respect to $\mathtt{ENV}''$ and ρ. We say that $(0, b'')$ is the *source* of (t', i') (at time 0) in $\mathtt{ENV}''$ if $(0, b'')$ is an ancestor of (t', i'), is final, and $\mathrm{dist}(b'', i') < \mathrm{dist}(j, i')$ for every other final $(0, j)$. Consider any feasible extension E'' of $\mathtt{ENV}_{t_1}[G]$. By Claim 11 and the discussion above, the source $(0, b'')$ of (t_1, g_2) (at time 0 in $\mathtt{ENV}''$) must satisfy $b'' \in [g_2 - t_1, g_2 - t_1 + \Delta - 1]$. Furthermore, $(0, b'')$ must also be the source of (t_1, g') for every grid location $g' \in [g_2 - t_1, g_2]$. Therefore, for each such grid location, $\mathtt{ENV}_{t_1}(g') = \mathtt{ENV}''_{t_1}(g') = f_\rho(\mathtt{ENV}''_0(b''), \mathsf{parity}(t_1), \mathsf{parity}(\mathrm{dist}(g', b'')))$, where $\mathtt{ENV}''_0(b'') = f_\rho^{\leftarrow}(\mathtt{ENV}_{t_1}(g_2), \mathsf{parity}(t_1), \mathsf{parity}(\mathrm{dist}(g_2, b'')))$. If f_ρ and $f_\rho^{\leftarrow}$ do no depend on their third argument, then, by Equation (7.3), $\mathtt{ENV}'_0(b') = \mathtt{ENV}''_0(b'')$ and if they do, then $\mathtt{ENV}'_0(b') = \mathtt{ENV}''_0(b'') \oplus \mathsf{parity}(\mathrm{dist}(b', b''))$. In either case, Equation (7.7) follows.

Pairs $(t, i) \in C$. By the definition of C, there exists a maximal $\overline{\mathcal{F}}_\rho$ grid interval $[g_1(i), g_2(i)]$ such that $g_1(i) \leq i \leq g_2(i)$. Additionally, the pair (t, i) does not descend from either the pair $(t_1, g_1(i) - 1)$ or from the pair $(t_1, g_2(i) + 1)$. Let $g(i)$ be the grid location defined in Definition 11 (of violating pairs in C), so that $g(i) \in G \cap [g_1(i), g_2(i)]$ and $\mathrm{dist}(i, g(i)) < \Delta$.

By the definition of $\mathtt{ENV}'_0$ (based on σ – recall Equation (7.1)), we have that $\mathtt{ENV}'_0(\Gamma_k(i)) = h^{\leftarrow}_{\rho}(\mathtt{ENV}_{t_1}(\Gamma_k(g(i))), \mathsf{parity}(t_1), \overrightarrow{\mathrm{dist}}(i, g(i)))$. By the definition of $h^{\leftarrow}_{\rho}$ (Definition 8), this implies that $\mathtt{ENV}_{t_1}(\Gamma_k(g(i))) = h_{\rho}(\mathtt{ENV}'_0(\Gamma_k(i)), \mathsf{parity}(t_1), \overrightarrow{\mathrm{dist}}(i, g(i)))$. Since $\mathtt{ENV}'_0(\Gamma_k(j)) \in \overline{\mathcal{F}}_{\rho}$ for every location $j \in J(g_1(i)), g_2(i))$, and the environment $\mathtt{ENV}'$ evolves according to ρ where ρ satisfies Condition 4, we know that $\mathtt{ENV}'_{t_1}(\Gamma_k(g(i))) = h_{\rho}(\mathtt{ENV}'_0(\Gamma_k(i)), \mathsf{parity}(t_1), \overrightarrow{\mathrm{dist}}(i, g(i)))$. Hence, $\mathtt{ENV}'_{t_1}(\Gamma_k(g(i))) = \mathtt{ENV}_{t_1}(\Gamma_k(g(i)))$. Furthermore, using in addition the fact that (t, i) does not descend from either $(t_1, g_1(i) - 1)$ or $(t_1, g_2(i) + 1)$, we get all ancestors of (t_1, j) of (t, i) satisfy $\mathtt{ENV}'_{t_1}(\Gamma_k(j)) \in \overline{\mathcal{F}}_{\rho}$, so that $\mathtt{ENV}'_t(\Gamma_k(i)) = h_{\rho}(\mathtt{ENV}'_{t_1}(\Gamma_k(g(i))), \mathsf{parity}(t - t_1), \overrightarrow{\mathrm{dist}}(g(i), i)))$. Since (t, i) is not a violating pair, $\mathtt{ENV}_t(\Gamma_k(i)) = h_{\rho}(\mathtt{ENV}_{t_1}(\Gamma_k(g(i))), \mathsf{parity}(t - t_1), \overrightarrow{\mathrm{dist}}(g(i), i))$, and using $\mathtt{ENV}'_{t_1}(\Gamma_k(g(i))) = \mathtt{ENV}_{t_1}(\Gamma_k(g(i)))$ we get that $\mathtt{ENV}'_t(i) = \mathtt{ENV}_t(i)$.

References

1 D. Burraston and E. Edmonds. Cellular automata in generative electronic music and sonic art: a historical and technical review. *Digital Creativity*, 16(3):165–185, 2005.

2 Bastien C. and Michel D. *Cellular Automata Modelling of Physical Systems.* Cambridge University Press, 1998.

3 P. P. Chaudhuri, D. R. Chowdhury, S. Nandi, and S. Chattopadhyay. *Additive Cellular Automata Theory and Applications. Vol. 1.* IEEE Press, 1997. IEEE Press advances in circuits and systems series.

4 M. Cook. Universality in elementary cellular automata. *Complex systems*, 15(1):1–40, 2004.

5 O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

6 O. Goldreich and D. Ron. On learning and testing dynamic environments. *Journal of the ACM*, 64(3):1–90, 2017.

7 L. Kier, C. Cheng, and P. Seybold. Cellular automata models of chemical systems. *SAR and QSAR in Environmental Research*, 11(2):79–102, 2000.

8 A. Mustafa, A. Heppenstall, H. Omrani, I. Saadi, M. Cools, and J. Teller. Modelling built-up expansion and densification with multinomial logistic regression, cellular automata and genetic algorithm. *Computers, Environment and Urban Systems*, 67:147–156, 2018.

9 Y. Nakar and D. Ron. Testing dynamic environments: Back to basics. *arXiv preprint*, 2021. arXiv:2105.00759.

10 D. Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science*, 282(2):231–257, 2002.

11 D. A Rosenblueth and C. Gershenson. A model of city traffic based on elementary cellular automata. *Complex Systems*, 19(4):305, 2011.

12 R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

13 J. von Neumann. The general and logical theory of automata. *Cerebral Mechanisms of Behavior: The Hixon Symposium*, pages 1–41, 1951.

14 S. Wolfram. *A new kind of science*, volume 5. Wolfram media Champaign, IL, 2002.

15 J. Xu, B. Gu, Y. Guo, J. Chang, Y. Ge, Y. Min, and X. Jin. A cellular automata model for population dynamics simulation of two plant species with different life strategies. In *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, 2010.

16 A. N. Zehmakan. *On the spread of information through graphs.* PhD thesis, ETH Zurich, 2019.

17 Z. Zheng, W. Huang, S. Li, and Y. Zeng. Forest fire spread simulating model using cellular automaton with extreme learning machine. *Ecological Modelling*, 348:33–43, 2017.